

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-34394

(P2001-34394A)

(43)公開日 平成13年2月9日(2001.2.9)

(51)Int.Cl. ⁷	識別記号	F I	テマコード [*] (参考)
G 0 6 F 3/02	3 2 0	G 0 6 F 3/02	3 2 0 A
	3 7 0		3 7 0 A
3/00	6 1 0	3/00	6 1 0
	6 5 4		6 5 4 A
17/22	5 0 2	17/22	5 0 2 A

審査請求 未請求 請求項の数19 OL (全 48 頁) 最終頁に続く

(21)出願番号 特願2000-160860(P2000-160860)

(22)出願日 平成12年5月30日(2000.5.30)

(31)優先権主張番号 1 3 6 8 0 3

(32)優先日 平成11年5月30日(1999.5.30)

(33)優先権主張国 米国 (U S)

(71)出願人 599011218

テジック・コミュニケーションズ・インコーポレーテッド

Tegic Communication
s Incorporatedアメリカ合衆国、ワシントン州 98121、
シアトル、ウエスタン・アベニュー 2001

(72)発明者 クリフォード・エー・クシュラー

アメリカ合衆国、ワシントン州 98036、
リンウッド、ラーチ・ウェイ・ナンバー
32、20715

(74)代理人 100058479

弁理士 鈴江 武彦 (外4名)

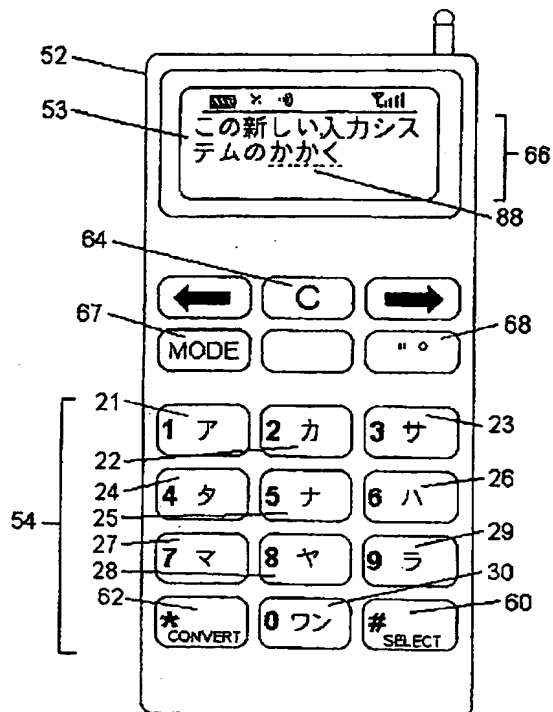
最終頁に続く

(54)【発明の名称】 音声仮名文字の入力順序を生成するための入力システム

(57)【要約】

【課題】 小型化されたキーボードによる入力システムを提供する。

【解決手段】 日本語用の小型化されたキーボードシステムであって、入力されたキーストロークの順序の単語レベルの曖昧さを使用し、そして、ユーザーにあいまいな入力キー順序の望ましい解釈を仮名として選択可能にし、そして、その選択された仮名解釈に関連付けられた望ましいテキストを選択させるシステムである。ここに開示されたものは、小型キーボードシステムであって、2つのキーストロークの順序を使用して各音節を特定するものである。その音節は口蓋音化され、2つの仮名を使用してあらわすものも含む。キーストロークの入力順序は、命令されたキーストロークのペアとして解釈され、それは2次元マトリックスにおける位置により1つの文字を選択する。



【特許請求の範囲】

【請求項 1】 ユーザーによって開始された、曖昧な入力シーケンスの曖昧さの解消のためと、日本語の原文出力の生成のためとの曖昧さ解消システムであって、この曖昧さ解消システムは、以下の構成要素 (a) から (d) までを有する：

(a) ユーザー入力装置は、複数の入力を有し、各入力が入力の複数の文字と関連しており、入力シーケンスは、ユーザー入力装置を操作することによって入力が選択されるたびに発生され、選択された入力のシーケンスに対応して発生された入力シーケンスは、各入力と関連付けられた複数の文字のために曖昧な原文解釈 (textual interpretation) を有している；

(b) メモリは、複数の読み方テキストオブジェクトを構成するために使用されるデータを有し、複数の読み方テキストオブジェクトの各々は、使用の頻度値と入力シーケンスとに関連づけられており、また、メモリ内の各読み方テキストオブジェクトは、ユーザーに出力される仮名のシーケンスを識別し、そして、前記読み方テキストオブジェクトは、ルートワードオブジェクトと、これらルートワードオブジェクトと対応した 1 もしくは複数の接尾辞オブジェクトから構成されている；

(c) ディスプレイを有している；

(d) プロセッサは、生成された入力シーケンスに従って、すべての読み方テキストオブジェクトをメモリから識別するように前記ユーザー入力装置と、メモリと、そしてディスプレイとに結合されており、前記識別された読み方テキストオブジェクトは、生成された入力シーケンスに関連しているルートワードオブジェクトであり、そして、前記識別された読み方テキストオブジェクトが、組み合わせが生成された入力シーケンスに関連づけられた時、接尾辞オブジェクトとルートワードオブジェクトの組み合わせであり、また、前記プロセッサは、生成された入力シーケンスの原文の解釈として、識別された読み方テキストオブジェクトの少なくとも 1 つをディスプレイに表示させる出力信号を生成させる音声仮名文字の入力順序を生成するための入力システム。

【請求項 2】 前記各接尾辞オブジェクトは、予め決定された接尾辞分類に関連づけられている、請求項 1 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 3】 前記ルートと接尾辞オブジェクトが、ツリー構造内のメモリに格納されており、このツリー構造は、複数のノードで構成されており、各ノードは、入力シーケンス及び 1 以上の読み方オブジェクトに関連づけられている、請求項 2 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 4】 各ルートワードオブジェクトは、これが少なくとも 1 つの接尾辞オブジェクトに関連づけられているとき、分類子から構成されており、この分類子は、

1 以上の予め定義された接尾辞分類を識別し、この接尾辞分類は、1 以上の接尾辞オブジェクトを有しており、この接尾辞オブジェクトは、ルートワードオブジェクトに関連づけられているように予め決定されている、請求項 3 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 5】 前記接尾辞オブジェクトは、各接尾辞分類内で、ツリー構造内のメモリに格納されており、また、前記ツリー構造は、複数のノードから構成されており、各ノードは、1 以上の読み方オブジェクト、及び入力シーケンスの少なくとも一部分に関連づけられており、そして、前記ルートワードオブジェクトは、ルートツリー構造のメモリに格納されており、また、前記ルートツリー構造は、複数のノードから構成されており、各ノードは、ルートワードオブジェクトと入力シーケンスの少なくとも一部分とに関連づけられていることを特徴とする、請求項 2 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 6】 各接尾辞オブジェクトは、予め定義された接尾辞分類に関連づけられている範囲内での他の接尾辞に対する接尾辞オブジェクトの使用の頻度を示すための使用値の頻度から構成されている、請求項 5 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 7】 前記相対的な使用値の頻度は、対数値である、請求項 6 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 8】 前記プロセッサは、関連した分類子と発生した入力シーケンスとに従った、すべての読み方テキストオブジェクトのメモリから判別する、請求項 6 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 9】 前記プロセッサは、識別された読み方テキストオブジェクトの各々の使用値の頻度の決定のための頻度要素から構成され、決定された使用値の頻度に従って、識別された読み方テキストオブジェクトの命令を決定し、決定された命令に従って、識別された読み方テキストオブジェクトをディスプレイに表示させるように命令する、請求項 8 記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項 10】 前記頻度構成要素は、以下に述べる (i) から (i i i) によって、識別された読み方テキストオブジェクトの各々の使用値の頻度を決定する、請求項 9 記載の音声仮名文字の入力順序を生成するための入力システム。

(i) 前記識別された読み方テキストオブジェクトが、全体の入力シーケンスと一致したルートワードオブジェクトであるとき、そして、このルートワードオブジェクトが、接尾辞を表すために予め定義されているとき、各識別された読み方テキストオブジェクトに関連づけられた使用頻度を決定する；

(i i) 前記識別された読み方テキストオブジェクトが、全体の入力シーケンスと一致したルートワードオブジェクトであるとき、そして、このルートワードオブジェクトが、接尾辞が付されていないことを表すために予め定義されているとき、予め定義されている相対頻度と各識別された読み方テキストオブジェクトに関連する使用値の頻度生産を決定する；

(i i i) 前記識別された読み方テキストオブジェクトが、1つまたはそれ以上の接尾辞オブジェクト、そして、ルートワードオブジェクトの組み合わせであるとき、前記1つまたはそれ以上の接尾辞オブジェクトに関連した使用値の相対頻度と、前記ルートワードオブジェクトに関連した使用値の頻度の生成を決定する。

【請求項11】 前記頻度構成要素は、前記決定された2以上の識別された読み方テキストオブジェクトの頻度が同じである場合、決定命令の構成要素の命令からさらに構成され、また、2以上の識別された読み方テキストオブジェクトが、同じ決定された使用値を持っている場合、前記命令構成要素は、より長いルートワードオブジェクトの識別された読み方テキストオブジェクトを識別し、そして、同じ決定された使用値の頻度の読み方テキストオブジェクトの別の識別された読み方テキストオブジェクトの前に、より長いルートワードオブジェクトの識別された読み方テキストオブジェクトを命令する；また、同じ決定された頻度の2以上の識別された読み方テキストオブジェクトが、同じ長さの異なるルートワードオブジェクトを持っている場合、ルートワードツリー構造での最初であるルートワードオブジェクトの識別された読み方テキストオブジェクトは、同じ長さのルートワードオブジェクトを持った、同じ決定された頻度の他の識別された読み方テキストオブジェクトより前に命令される；そして、同じ決定された頻度の2以上の識別された読み方テキストオブジェクトが、同じルートワードオブジェクトを持っている場合、接尾辞ツリー構造で最初であるルート接尾辞オブジェクトの識別された読み方テキストオブジェクトは、同じルートワードオブジェクトを持っている同じ決定された頻度の他の識別された読み方テキストオブジェクトより前に命令される請求項10記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項12】 前記接尾辞分類は、1以上の動詞接尾辞分類のセットと、1つまたはそれ以上の名詞接尾辞分類のセットと、及び、1つまたはそれ以上の形容詞接尾辞分類のセットとに分類されていることを特徴とする、請求項2記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項13】 前記接尾辞オブジェクトは、動詞、名詞及び形容詞接尾辞分類のセットの範囲内で、ツリー構造内のメモリに格納されており、前記ツリー構造は、複数のノードから構成されており、各ノードは、1以上の

読み方オブジェクトと、少なくとも入力シーケンスの一部とに関連づけられており、そして、前記ルートワードオブジェクトは、ルートツリー構造内のメモリに格納されており、前記ルートツリー構造は、複数のノードから構成されており、各ノードは、入力シーケンスとルートワードオブジェクトの少なくとも一部分に関連づけられている、請求項12記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項14】 各接尾辞オブジェクトは、接尾辞分類の範囲内で、他の接尾辞オブジェクトに関係した、接尾辞オブジェクトの使用値の頻度を命令するための使用値の頻度から構成されている、請求項13記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項15】 前記接尾辞オブジェクトと関連する使用値の相対頻度が対数関数的な値である、請求項14記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項16】 前記プロセッサがメモリから生成された入力シーケンスおよび付随する分類命令子による全ての読み方テキストオブジェクトを識別する、請求項3記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項17】 前記プロセッサが、各々の識別された読み方テキストオブジェクトの使用値の頻度を決定して、使用値の決定された頻度による識別された読み方テキストオブジェクトの順序を決定して、ディスプレイに決定された順序による識別された読み方テキストオブジェクトを表示するように命令するための頻度構成要素から構成されている、請求項16記載の音声仮名文字の入力順序を生成するための入力システム。

【請求項18】 前記頻度構成要素が、以下に述べる(i)から(i i i)によって、各々の識別された読み方テキストオブジェクトの使用値の頻度を決定する、請求項17記載の音声仮名文字の入力順序を生成するための入力システム。

(i) 識別された読み方テキストオブジェクトが、全体の入力シーケンスに一致したルートワードオブジェクトである場合、そして、前記ルートワードオブジェクトが接尾辞を付されて表されるために予め定義されている場合、各識別された読み方テキストオブジェクトに関連づけられた使用頻度を決定する；

(i i) 識別された読み方テキストオブジェクトが、全体の入力シーケンスと一致するルートワードオブジェクトである場合、そして、前記ルートワードオブジェクトが、接尾辞が付されずに表されるために予め定義されている場合、予め定義された相対頻度と、各識別された読み方テキストオブジェクトに関連づけられた使用値の頻度の生成を決定する；

(i i i) 識別された読み方テキストオブジェクトが、ルートワードオブジェクトと1つまたはそれ以上の接

10

20

30

40

50

尾辞オブジェクトとの組み合わせであった場合、前記 1 またはそれ以上の接尾辞オブジェクトに関連づけられた使用値の相対頻度と、前記ルートワードオブジェクトに関連づけられた使用値の頻度の生成を決定する。

【請求項 19】 前記頻度構成要素は、少なくとも 2 つの識別された頻度が、読み方テキストオブジェクトを識別した順序が同じものであると決定するための配列構成要素から更に構成され、また、少なくとも 2 つの識別された読み方テキストオブジェクトが使用値の同じ決定された頻度を有する時、もう一方が使用値の同じ決定された頻度を有する読み方テキストオブジェクトの読み方テキストオブジェクトを識別する前に、前記配列構成要素は、識別された読み方テキストオブジェクトをより長いルートワードオブジェクトと同一視して、識別された読み方により長いルートワードオブジェクトを有するテキストオブジェクトを命じる；そして、同じ決定された頻度を有する少なくとも 2 つの識別された読み方テキストオブジェクトが、同じ長さの異なるルートワードオブジェクトを有するときに、ルートワードツリー構造において、最初のワードオブジェクトが同じ決定された頻度を有する他の識別された読み方テキストオブジェクトの前に、注文されるルートワードを有する識別された読み方テキストオブジェクトが同じ長さのルートワードオブジェクトを有する；また、同じ決定された頻度を有する少なくとも 2 つの識別された読み方テキストオブジェクトが同じルートワードオブジェクトを有するときに、接尾辞ツリー構造において最初のルート接尾辞オブジェクトを有する識別された読み方テキストオブジェクトは、同じルートワードオブジェクトを有する同じ決定された頻度を有する他の識別された読み方テキストオブジェクトの前に命じられる、請求項 18 記載の音声仮名文字の入力順序を生成するための入力システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は小型化されたキーボードシステム、特に日本語の平仮名、カタカナ、漢字文字からなるテキストを作成するための小型化されたキーボードシステムに関する。

【0002】

【従来の技術】多年にわたって、ポータブルコンピュータはますます小型化されている。より小さいポータブルコンピュータを製造するための主要なサイズ限定構成要素はキーボードである。標準的なタイプライターサイズのキーが使用されるならば、ポータブルコンピュータは少なくともキーボード程の大きさでなければならない。小型化されたキーボードがポータブルコンピュータで使用されているが、小型化されたキーボードのキーは非常に小さいのでユーザーが容易にまたは迅速に操作できないことが知られている。

【0003】ポータブルコンピュータにフルサイズのキ

ーボードを設けることもコンピュータの真のポータブルな使用を妨げる。大部分のポータブルコンピュータはユーザーが両手でタイプすることができるようコンピュータを平坦な作業面に配置せずには動作できない。ユーザーは立ちながらまたは移動しながらポータブルコンピュータを容易に使用することはできない。パーソナルデジタルアシスタント (PDA) または手のひらサイズのコンピュータと呼ばれる最新の世代の小型ポータブルコンピュータにおいては、企業は装置に手書き認識ソフトウェアを内蔵することによってこの問題を解決しようとしている。ユーザーはタッチ感応パネルまたはスクリーン上に書き込むことによって直接テキストを入力してもよい。この手書きテキストは認識ソフトウェアによってデジタルデータに変換される。残念ながら、ペンによる印刷または書き込みは一般的にタイピングよりも遅いことに加えて、手書き認識ソフトウェアの正確性および速度は現在に至るまで適切ではない。多数の複雑な文字が存在する日本語の場合、この問題は特に困難になる。さらに悪いことに、テキスト入力が必要とする今日の持ち型のコンピュータ装置はさらに小型になっている。最近の両方向ページング、セルラ電話、その他のポータブル無線技術の進展によって、小型のポータブル両方向メッセージシステム、特に電子メール (e-メール) の送信と受信との両者を行うことができるシステムが要求される。

【0004】それ故、小型であり、ユーザーが一方の手で装置を保持しながら他方の手で動作可能であるコンピュータ装置へテキストを入力するためのキーボードを開発することが有効である。従来の開発はキーの数を減少しているキーボードの使用を考慮している。タッチトーン電話のキーパッドレイアウトにより示唆されているように、多数の減少された数のキーボードは 3×4 アレイのキーを使用している。アレイ中のキー数は多数の文字と関連されている。それ故、ユーザーが所定のキーに関連する文字のうちのいずれが所望の文字であるかを命令する方法が必要とされる。

【0005】減少されたキー数のキーボードに入力された平仮名文字を曖昧性がなく特定する 1 つの提案された方法はユーザーが各仮名を特定するのに 2 以上のキーストロークを入力することを必要とする。キーストロークは同時 (コーディング) または順次連続的に (多ストローク仕様) 入力されてもよい。コーディングによっても多ストローク仕様によっても、キーボードが使用上の適切な簡潔性および効率を有することはできない。多ストローク仕様は非効率であり、コーディングはしばしば学習および使用に複雑である。

【0006】日本語の 50 音図表の各音節は、1 つの母音または母音が後続する子音からなる。2 つの例外があり、即ち音節「ん」は母音がなく、小さい「っ」は後に続く子音の発音の「二重化」または「硬音化」を示すこ

とに使用される。これらの音節は（日本語固有の単語を書くときに共通して使用される）平仮名、または（外来語の単語を書くときに通常使用される）カタカナとして書かれる。用語仮名は、平仮名またはカタカナのいずれか一方を表すのに使用される。音節は通常（図35で示されている）行または列の表として表されており、ここで各行は5つの日本語の母音「あ」、「い」、「う」、「え」、「お」に対応して5つまでの列のエントリを有する。所定の子音は行内においてあるエントリの音声変化を受けるが（例えば s (a) さ→sh (i) し; t (a) た→ts (u) つ; 等）、各行は最初の子音に対応する。第1の行は最初の子音のない各5つの母音に対応する5つの音節からなる。8番目の行は口蓋音化された母音「や」、「ゆ」、「よ」からなる（YIとYEは現代日本語では使用されていない）。区分発音符「[˙]」と「[˚]」は子音の発音変化を示すために使用され、通常無声音を有声の子音に変化することを示している。図36は図35の音節に区分発音符「[˙]」と「[˚]」を付加することによって形成された音節を示している。音節「や」、「ゆ」、「よ」の小さいバージョンも、対応する子音と口蓋音化された母音からなる音節（例えば「きゃ」を表示するために「き」に、小さい「や」を後続）を表すために図35および2の第2の行または「い」の列の音節と組み合わせて使用されることができ、口蓋音化された母音を有するこれらの音節はしたがって図37で示されているように1対の仮名として書かれ、これは区分発音符と共に書かれる形態を含んでいる。

【0007】日本語の辞書編集順序は通常図35の（母音Aに対応する）第1の列、即ち「あ」、「か」、「さ」、「た」、「な」、「は」、「ま」、「や」、「ら」、「わ」、「ん」の音節の順序（順序）により表され、ここでこれらの各音節（特定の発音のものを除く）はその順序の母音「あ」、「い」、「う」、「え」、「お」から構成される5つまでの音節のサブ分類を表している。現在、日本語のテキスト入力を要求するセルラ電話等の製品は通常、多ストローク仕様方法を使用し、ここでは9個のキーのそれぞれが各第1の9つの行（「あ」から「ら」）に関連する。多数のキーストロークは、対応する行の音節のうちのどの音節を意図しているかを示すために使用され、ここでは各付加的なキーストロークは逐次的に文字に変化して図35または2の次の列で現れる文字に出力される。キー分離またはタイムアウト方法が同一のキーに関連する連続的な符号の入力を可能にすることに使用される。10番目のキーが音節「わ」、「を」、「ん」と、先行する音節の母音を反復する母音のみの音節を示しているカタカナの「b o」符号に使用される。小さい「や」、「ゆ」、「よ」、も「や」のキーに関連され、付加的なキーストロークの選択を必要とする。付加的なキーは共通して音節の後に区分発音符を付加するために使用される。

【0008】小型化されたキーボードを使用した日本語の平仮名（またはカタカナ）の入力は挑戦的な問題であり続けている。前述した現在の多ストローク方法により、1つの仮名音節の生成は平均して少なくとも3つのキーストロークを必要とする。2つの文字（即ち、小さい「ゃ」、「ゅ」、「ょ」が後続する図35および2の第2の列、または「い」列）により表されている口蓋音化された母音を有する音節は8つまでのキーストロークを行うことを必要とする。それ故、平仮名の入力に必要とするキーストロークの数を最少にし使用が簡単で直観的な小型化されたキーボードシステムを開発することが望まれる。

【0009】仮名に加えて中国文字（漢字）を含む標準的な日本語テキストのタイピングはより一層挑戦的な問題である。全てのキーボードと大きなディスプレイを有する標準的なコンピュータにおけるテキストの入力は通常、図35-31で示されているように各平仮名音節に対応する（日本語で「ローマ字」と呼ばれる）ラテンアルファベットの文字を使用して最初に所望のテキストの発音をタイプすることによって実現される。文字がタイプされるとき、入力は自動的に対応する平仮名音節に変換されスクリーン上で表示される。多くの場合、ユーザーは、平仮名として最初に表示されたテキストを所望される特定の原文解釈に変換することを必要とする。表示された平仮名は、ユーザーが実際に入力したい漢字と平仮名を組合わせた音声的読みを表し、これはユーザーが意図している意味を伝達するものである。日本語の同音異義語が多数であるため、ユーザーにより入力された平仮名に対応する漢字と平仮名の意味を有する組合わせは多数である。標準的なコンピュータでは、これらの多数の代わりの変換が表示されることができ、ここでは例えば、それぞれの代わりの変換が数字キーに関連され、それによってキーストロークは入力された平仮名を表示される漢字解釈へ変換する。限定されたディスプレイサイズと使用可能なキーが少数であるために、小型のハンドヘルド装置にこのプロセスを構成しようとするときに付加的な複雑さが生じる。

【0010】小型化されたキーボードに入力された平仮名を特定する代わりの方法はユーザーが各平仮名を1つのキーストロークで入力することを可能にする。小型化されたキーボードの各キーは多数の平仮名文字に関連されている。ユーザーが連続したキー入力をするとき、各キーストロークはいくつかの平仮名のうちの1つを示しているので結果的な出力に曖昧さが存在する。それ故、システムは、ユーザーが各キーストロークの可能な解釈のうち目的とするものを実効的に示すことができる手段を提供しなければならない。キーストローク順序の曖昧さを解決する幾つかの方法が提案されている。

【0011】曖昧なキーストローク順序に対応する正確な文字順序を決定する多数の提案された方法はJohn L.

ArnettとMuhammad Y. Javadによる文献 (Probabilistic Character Disambiguation for Reduced Keyboards Using Small Text Samples” Journal of the International Society for Augmentative and Alternative Communication (以後「アノットの文献」とする)) に要約されている。アノットの文献は、大多数の曖昧化をなくす方法が所定の文脈における文字の曖昧さを解決するため既知の関連言語の文字順序の統計を使用していることに留意している。即ち、既存の曖昧化をなくすシステムは、キーストロークの適切な解釈を決定するために明瞭なキーストロークグループがユーザーにより入力されるとき、これらを統計的に解析する。アノットの文献はまた小型化されたキーボードからテキストを復号するため幾つかの曖昧化をなくすシステムが単語レベルの明瞭さの使用しようとしていることにも留意している。単語レベルの曖昧化をなくすプロセスは、可能な語尾を意味する明瞭な符号 (文字) を受信した後、受信されたキーストロークの順序全体を辞書との可能な一致を比較することによって単語を完成する。アノットの文献は多数の単語レベルの曖昧さをなくす技術の欠点を論述している。例えば、珍しい単語を識別する制限と、辞書にない単語を復号する能力がないため、単語レベルの不明瞭さはしばしば正確に単語を復号できない。復号の制限のために、単語レベルの曖昧さをなくすには、1文字当たり1回のキーストロークの効率である無制限の英語テキストにおけるエラーのない復号を与えない。それ故、アノットの文献は単語レベルの曖昧さをなくすよりも文字レベルの曖昧さをなくすことに集中し、文字レベルの曖昧さをなくすことが最も有望な曖昧化をなくす技術であるように思われることを示している。しかしながら、アルファベット言語と対照的に、日本語の平仮名文字は基本的に音素であることを表す1文字ではなく音節を表している。この理由で、平仮名の可能な順序にほとんど制限がないので文字レベルの明瞭さは日本語では非効率であり、平仮名順序の確率分布はこの方法では十分実効的に適合されない。

【0012】さらに別の提案された方法はI. H. Wittenによるテキストブック (“Principle of Computer Speech”, Academic Press, 1982年 (以後「ウィテン方法」とする)) に開示されている。ウィテンは24, 500語の英語辞書の約92%において、キーストローク順序と辞書とを比較したときに曖昧性が生じないことを認識している。しかしながら、曖昧さが生じるとき、ウィテンはユーザーに曖昧性を示し複数の曖昧性のある入力で選択を行うようにユーザーに要求するシステムによってこれらが相互に解決されなければならないことに留意している。それ故、ユーザーは各語尾でシステムの予測に回答しなければならない。このような回答はシステムの効率を遅くし、所定のテキストセグメントの入力に必要なキーストローク数を増加する。

【0013】日本語の場合、ワードプロセッサソフトウェアのユーザーは言語に多数の同音異義語が存在するため、多数の曖昧な解釈から後に続く単語入力を選択しなければならない。仮名の同一順序が頻繁に2以上の異なった漢字解釈へ変換されることができる。したがって、仮名の順序の入力後、ユーザーは通常、1組の可能な選択から所望の漢字変換を選択することを要求され、しばしば正確な変換が選択されているかを確証することも必要とされる。小型化されたキーボードを使用して平仮名が入力されたとき、漢字に変換される平仮名の順序としてユーザーが実際に意図しているものに関する曖昧さが存在する。結果として、可能な解釈数は非常に増加される。

【0014】曖昧なキーストローク順序の曖昧化をなくすされることが挑戦的な問題として続けられている。前述の文献で注目されていたように、テキストのセグメントの入力に必要なキーストローク数を最小化する満足すべき解決策は、ポータブルコンピュータで使用するのに許容可能であるような必要な効率を得ることはできない。それ故、必要なキーストロークの総数を最少にしながら、簡単に、ユーザーインターフェイスの容易な文脈内に入力されるキーストロークの曖昧さを解決する曖昧化をなくすシステムを開発することが望ましい。このようなシステムはそれによってテキスト入力の効率を最大にする。

【0015】日本語の実効的な小型化キーボード入力システムは以下の全ての基準を満たさなければならない。第1に、キーボード上の日本語の音節 (仮名) の配列と、これらが生成される方法とは、日本人の話者に対して使用法の理解と学習が容易でなければならない。第2に、システムはテキストの入力に必要なキーストローク数を最少にし、それによって小型化されたキーボードシステムの効率を強化しなければならない。第3に、システムは入力プロセス中に必要な注意および決定を行う量を減少することによってユーザーの認識負担を減少しなければならない。第4に、この方法は実効的なシステムを構成するのに必要なメモリと処理リソース量を最少にすべきである。

【0016】Kisaichi等 [JP 8-314920; US 5, 786, 776; EP 0 732 646A2] は電話のキーパッドのキー1-0がそれぞれ平仮名音節 {あいうえお}、{かきくけこ}、{さしすせそ}、{たちつと}、{なにぬねの}、{はひふへほ}、{まみむめも}、{やゆよ}、{らりるれろ}、{わをん} というラベルを付けられている方法を開示している。これは電話のキーパッドのキー1-9がそれぞれ平仮名音節「あ」、「か」、「さ」、「た」、「な」、「は」、「ま」、「や」、「ら」とラベルを付けられている日本の電話キーパッドのデファクトスタンダードに対応している。各キー上に表されている1つの平仮名は、1つの

平仮名が第1の列に表示されている図35で見られる平仮名の行全体に対応して、そのキーに割当てられた完全なセットの平仮名を表している。0キーはしばしば空白に{わをん}とラベルを付けられる。

【0017】Kisaichi等は単語レベルの曖昧化をなくす方法を開示しており、ここでユーザーは各文字が1回(時間)に関連しているキーを押すことによって文字(平仮名)の順序を曖昧に入力する。各入力順序の終わりにおいて、ユーザーは「変換/次候補」キーを押し、入力キー順序に関する平仮名の可能な順序のうち1つの第1の逐語解釈を表示する。Kisaichiは辞書構造を説明し、ここでは所定の入力キー順序の全ての逐語解釈がメモリの近接するブロックに連続して記憶されている。

「変換/次候補」キーを付加的に押すと、キーは辞書に記憶されている次の逐語解釈を、それが存在するならば表示する。逐語解釈が存在しないならば、エラーメッセージが表示され、任意選択的な変則処理が実行されてもよい。所望の逐語解釈が表示されたとき、ユーザーが次のテキストオブジェクトを入力し続けることができる前に所望のテキストが表示されていることを確認するため特別な「確定」キーが押されなければならない。

【0018】Kisaichi等により説明された方法には多くの難点がある。その1つは平仮名ストリングの特定と、各可能な平仮名候補ストリングの変換との両者に曖昧性が存在する事実により、所定のキー順序に多数の可能な逐語解釈が存在しがちであることである。これは所望の解釈を見つけ出すためにユーザーが「変換/次候補」キーを使用して多数の解釈を経過するステップを必要とする。さらに、可能な解釈のステップを通じて、ユーザーは入力の曖昧さによる種々の平仮名ストリングに対応する種々の漢字および/または平仮名ストリングを観察する。これは面倒であり、所望の解釈を見つけようとするユーザーに付加的な注意を要求する。さらに、逐語解釈のデータベースは、全てのデータが完全な単語のみからなり、所定の長さの全てのキー順序の全てのデータもメモリの近接ブロックに連続的に記憶されている。Kisaichi等は、任意の完全な単語に対応しない入力順序のこれらの点において、長く、しかも不完全な単語に対応する適切なシステムを表示することを可能にする方法を説明していない。入力中のこれらの点で、Kisaichi等のシステムは数字またはデフォルト文字または符号等の入力された各キーのデフォルト命令を表示するだけである。これはユーザーを混乱させ、目的とするキーが入力されていることをユーザーが確認するのに実効的なフィードバックを与えない。最終的に、ユーザーは各単語入力用の「確定」キーを押すことを必要とされ、各入力に付加的なキーストロークを入力しなければならない。したがってKisaichiにより開示されたシステムは前述の基準を満たさない。

【0019】単語レベルの曖昧化をなくすアプリケーシ

ョンに直面した別の重要な挑戦は、使用するのに最も便利な一種のソフトウェアプラットフォームでこれを適切に実行することである。前述したように、このような装置は両方向ページャ、セルラ電話およびその他のハンドヘルド無線通信装置を含んでいる。これらのシステムはバッテリー駆動であり、結果的にソフトウェア設計とリソース利用においてできる限り節約的に設計されている。このようなシステムで動作するように設計されたアプリケーションはプロセッサ帯域幅使用とメモリの必要性との両者を最少にしなければならない。これらの2つの要素は通常、相反して関連しがちである。単語レベルの曖昧化をなくすシステムは単語の大きなデータベースが機能することを必要とし、適切なユーザーインターフェイスを与えるために入力キーストロークに迅速にตอบสนองしなければならないので、使用するのに必要な処理時間に大きな影響を与えずに必要なデータベースを圧縮できることが非常に有効である。日本語の場合、仮名の順序をユーザーが目的とする漢字に変換することを支援するために付加的な情報がデータベースに含まなければならない。

【0020】単語レベルの曖昧化をなくすアプリケーションに直面した別の重要な挑戦は、入力されるキーストロークについてユーザーに十分なフィードバックを与えることである。一般的なタイプライターまたはワープロにより、各キーストロークは入力してすぐにユーザーに対して表示されることが特定の文字を表す。しかし、単語レベルの曖昧化をなくすには、各キーストロークは多数の文字を表し、キーストロークの任意の順序は多数の単語または単語ステムに一致するので、これはしばしば可能ではない。それ故、入力されたキーストロークの曖昧さを最少にし、テキスト入力中に生じる曖昧さをユーザーが解決できる効率を最大にする曖昧化をなくすシステムを開発することが所望される。ユーザーの効率を増加する1方法は各キーストロークについての適切なフィードバックを提供することであり、これは各キーストロークについての最も可能性の高い単語を表示することを含んでおり、現在のキーストローク順序が完全な単語に対応しない場合、まだ完成されていない単語の最も可能性の高いステムを表示する。

【0021】

【発明が解決しようとする課題】日本語の実効的な小型化されたキーボード入力システムを製造するため、前述の全ての基準を満たすシステムが設計されなければならない。第1に、キーボード上における日本語(仮名)の音節の配列と、これらが生成される方法は日本人がその使用を理解し学習することが容易であることが必要である。第2に、システムはテキストの入力に必要なとされるキーストローク数を最少にすることが必要である。第3に、システムは入力プロセス中に必要な注意および決定を行う量を減少し、適切なフィードバックを与えること

によってユーザーの認識負担を減少することが必要である。第4に、ここで説明した方法は実際のシステムを構成するのに必要なメモリと処理リソース量を最少にすることが必要である。

【0022】

【課題を解決するための手段】本発明は、単語レベルの曖昧化をなくす技術を使用して小型化されたキーボードを与え、それによってキーストロークの曖昧さを解決し日本語でテキストを入力する。キーボードは標準的な電話キーパッド上のようにフルサイズの機械キー、好ましくは3列×4行に整列された12のキーで構成されてよい。代わりに、キーボードはタッチ感度のディスプレイパネルに構成されることができ、ここではディスプレイ表面との接触により、接触位置に対応してシステムに入力信号が発生される。

【0023】複数の仮名文字または符号は少なくとも幾つかの各キーに割当てられており、それによってユーザーによるキーストロークは曖昧である。ユーザーはキーストローク順序を入力し、各キーストロークは1文字の仮名の入力を目的とする。したがって各キーストローク順序は単語または共通の語句の音声表示（以下、日本語の「読み方」とする）を表すことを目的とする。個々のキーストロークは曖昧であるので、キーストローク順序は潜在的に1以上の読み方を同数の仮名に一致する。

【0024】順序に対応する読み方と整合するために、キーストローク順序は1以上の記憶された語彙モジュールと比較することによって処理される。所定のキー順序に関連する種々の読み方は、一般的な使用における予測された使用頻度で決定される順序で語彙モジュールに記憶されており、読み方の予測された頻度は一般的な使用

（以下、日本語の「見出し語」とする）におけるその読み方（漢字、平仮名、カタカナまたはその組合わせからなる単語を含む）の全ての可能な逐語解釈の使用頻度の合計として計算される。別の好ましい実施形態では、読み方と見出し語は一般的な使用におけるそれらの予測された発生頻度により決定される順序で最初に記憶され、この順序はシステムユーザーによる実際の使用頻度を反映するように変化される。キーストロークの順序に一致し少なくとも1つの完全な単語または語句に対応している最も頻度の高い読み方は自動的に識別され、各キーストロークが受信されたときにディスプレイ上でユーザーに示される。読み方がキーストロークの順序に一致しない不完全な単語または語句が存在したならば、まだ完全でない単語または語句の最も共通して使用されるキーストロークが自動的に識別されディスプレイ上でユーザーに示される。用語「選択リスト」は入力キーストローク順序に対応して、システムによって発生された逐語解釈（読み方または見出し語）の任意のリストを意味することに一般的に使用される。使用可能な十分なディスプレイ領域を有する装置（以下「大型スクリーン装置」

と呼ぶ) 上で、選択リストはディスプレイの「選択リスト領域」に（全体的または部分的に）示されてもよい。このような装置上で、各キーストロークが受信されたとき、入力順序に対応する種々の読み方は、選択リスト領域の予測された頻度の高いものから低いものへの順序で、ディスプレイ上のリストでユーザーに同時に示される。限定されたディスプレイ領域を有する装置上で、選択リストは内部に維持され、リストのテキストオブジェクトは以下説明する選択キーの付勢に応答して一度に一つづつ表示される。1以上の代わりの逐語解釈に関連する記憶された語彙モジュールの各読み方または見出し語では、見出し語は一般的な使用において使用の予測頻度の減少する順序で記憶されており、それによって最も共通して使用される見出し語が最初に示される。簡単に前述したように、代わりの実施形態では、システムはユーザーが最も頻繁に出力するように選択した見出し語の記憶を残し、最も頻繁に選択される見出し語を最初に示すために表示の順序を変更する。

【0025】本発明の1観点によれば、ユーザーは曖昧性のない選択キーを押し、入力されたキーストローク順序を定める。選択キーの受信後、曖昧化をなくすシステムは最も頻繁に発生される読み方を自動的に選択し、ユーザーが付加的なテキストを入力し続けるときに構成されたテキストに仮名を付加する。デフォルトにより、仮名が対応する平仮名がない幾つかのカタカナのうちの1つでない場合（例えば「ヴ」）を除いて、読み方は平仮名の形態でディスプレイに示される。別の実施形態では、読み方がカタカナの形態、または後述の幾つかの代わりのキーボード配列、ローマ字の形態で表示される。

【0026】本発明の別の観点によれば、キーストローク順序の末尾を定めるために押された選択キーはまた普通に発生することが少ない読み方を選択するためにも使用される。ディスプレイで示される最も普通に使用される読み方が所望の読み方ではないならば、ユーザーは最も頻度の高い読み方から次に使用頻度が高い読み方へ移行するように再度選択キーを押し、これにより最初に表示された読み方を置換する。これが所望の読み方ではないならば、ユーザーは再度選択キーを押し、三番目に使用頻度の高い読み方へ移行する。選択キーを反復的に押すことによって、ユーザーは記憶された語彙モジュールから所望の読み方を選択する。本発明の別の観点によれば、記憶された語彙に見られる最後の読み方に到達したとき、第1の最も頻度の高い読み方が再度表示され、このサイクルが反復される。

【0027】本発明のさらに別の観点によれば、入力順序の各キーストロークはまたそのキーに関連する数字として解釈され、それによって関連する読み方について表示された最後のアイテムは入力キー順序に対応する数である。この数は出力に対して選択され、したがってシステムにおける別々の数字モードの必要性をなくす。

【0028】本発明の別の観点によれば、ユーザーが入力キー順序に対応する所望の読み方を一度選択すると、所望の見出し語は選択されている読み方と同一であり、仮名テキストとして既にディスプレイ上に表示されているならば（即ち変換が必要とされない）、ユーザーは入力される次の所望のテキストに対応するキーを単に押し続ける。付加的な確定または変換キーストロークは必要とされず、選択されたテキストが出力のためにディスプレイに既に一時的に送信され、これが（例えば選択キーを付加的に押すことによって）明白に変更されていないならば、出力テキストの一部として残る。ユーザーはすぐに付加的な後に続くテキストを入力し続け、テキストカーソル位置を移動するかまたは幾つかのその他のシステム機能を選択できる。所望の見出し語がディスプレイ上の選択された読み方ではないならば、即ち所望の見出し語は漢字、漢字+平仮名またはカタカナからなるならば、ユーザーは所望の見出し語が表示されるまで変換キーを押す。キー順序の入力後に表示される第1（デフォルト）の読み方が所望の読み方であるならば、ユーザーは選択キーを押す必要がなく、所望の見出し語を得るために直ちに交換キーを押してもよい。

【0029】本発明は前述の2つの基準を満たすような方法で、小型化されたキーボード上で順序が定められたキーストローク対から日本語の音節を明瞭に生成する方法を開示している。第1に、キーボード上における日本語（仮名）の音節の配列と、これらが生成される方法は日本人がその使用を理解し学習することが容易である。第2に、この配列は、日本語の音節を曖昧さがなく入力するのに必要なキーストローク数を最少にする。本発明のこの観点では、2つのキーストロークの順序は、それぞれ2つの仮名で書かれている図37で示された口蓋音の母音を有する音節を含む各音節を明瞭に特定するために入力される。

【0030】キーストロークの入力順序は、2次元マトリックスの位置にしたがって文字を選択する順序を定められたキーストローク対として解釈される。それぞれの順序を定められた対の第1のキーストロークは所望の文字が現れているマトリックスの行を特定し、各対の第2のキーストロークは列を特定する。マトリックスの第1の5つの列の文字の組織は、図35で示されているように日本語の50音表が日本人の話者により学習され概念化されている方法に準じている。基本的な50音表（図37）から別々のマトリックスとして通常考慮されているが、口蓋音の母音を有する音節（それぞれ2つの仮名の組合わせ）が形成される態様の自然なモデルに対応する方法で、付加的な2つの列が組織されてもよい。第1の8つの列の簡単なパターンに適合しない2つの特別なケース（小さい「っ」と「ん」）を処理するためにさらに2つの列が付加されてもよい。これらの2つの文字は種々の別の実施形態でやや異なって管理されることもで

きる。このマトリックス方法の簡潔さと論理的組織は、フィードバックをユーザーに与えるために使用可能な表示がないときでさえもマトリックスを使用することを可能にする。ディスプレイが使用可能であるとき、マトリックスは、ユーザーに対して透明なシステム動作を行うようにフィードバックが行われる態様を組織するために使用されることができる。

【0031】日本語の50音表は108の音節（小さい「っ」、「ゃ」、「ゅ」、「ょ」は、これらが異なった方法で書かれ発音されるので全角の「っ」、「ゃ」、「ゅ」、「ょ」とは別の音節として数える）を含んでいる。主としてカタカナで使用される母音音節の「あ」、「い」、「う」、「え」、「お」の小さい形態の音節等、使用の少ない幾つかの付加的な音節が存在する。これらの使用の少ない音節も前述したようにマトリックスシステムを使用して容易に生成されてもよい。108の音節のうち37は区分発音符号である濁点（゛）または半濁点（゜）のうちの一方を他の71の音節に単に付加することによって生成される。区分発音符号のないこれらの71の音節は9または10の行と8乃至10の列の単一マトリックスへ論理的に組織されることができる。本発明の小型化されたキーボード上の複数のキーは2つの仮名でラベルを付けられ、一方は所定のマトリックス行に関する子音を表し、第2の仮名は所定のマトリックス列に関する母音を表している。

【0032】この組織は108のうち106の音節において論理的で日本人の話者に直観的であり、残りの2つの音節、即ち小さい「っ」と「ん」を生成する方法は簡単で学習しやすい。2つの別々の仮名により表される口蓋音の母音を有する音節を含んだあらゆる音節は1対のキーストロークにより生成される。これによって仮名を入力するのに現在使用されている多キーストローク方法で必要とされるよりも小型化されたキーボードでキーストローク数も非常に少なくすることができる。したがって本発明は日本人が理解しやすく迅速に学習できる小型化されたキーボードを提供し、これは入力キーストローク順序の長さを減少することに関して有効である。

【0033】本発明の別の観点では、前述したように音節を特定する曖昧な方法または明瞭な方法との両者は、入力方法でより大きな実効性を実現するために組合わされてもよい。好ましい1実施形態では、生成される各単語または語句の第1の音節は前述したようにマトリックス方法を使用して順序を定められた1対のキーストロークを入力することによって明瞭に特定化され、単語または語句の残りの音節は単語レベルの曖昧化をなくす方法を使用して各音節で1つのキーストロークにより曖昧さを有して特定される。

【0034】本発明のさらに別の観点では、キーストローク順序の多数の解釈が選択リストでユーザーへ与えられる。キーストローク順序は1以上の単語を形成するも

のとして解釈され、ここでは最も使用頻度の高い対応する単語が表示され、他の対応する単語は選択リストに表示されてもよい。同時にキーストローク順序は、数（前述したように）として、本発明の2ストローク方法またはよく知られている多ストローク特定方法を使用して入力された単語として、および不完全な単語のステムとして解釈される。大型スクリーン装置上では、多数の解釈が同時に、ユーザーにより入力されたキーストローク順序の各キーストロークの選択リスト領域でユーザーへ示される。任意の装置上で、ユーザーは選択キーを複数回押すことによって別の解釈（変換候補）から選択してもよい。

【0035】本発明のさらに別の観点では、ユーザーは、単語再変換モードになることによって、すでに完全なキー入力されたものの曖昧さを明白になくす。再変換モードにおいて、ディジットキー入力は、2ストロークの明白な入力の第2のキー入力である（第1のキー入力は、キー入力の存在している中においてキー入力が存在している）。キー入力で与えられたメニューは、現在の文字位置と生成する文字とのために有効であり、自動的に表示される。カーソルは、各キー入力による次の文字入力を自動的に促進させる。選択リストは、新しい明白な文字による目的の語の適合のみを表すために簡潔になっており、継続的に最新のものに更新され、表示される。このモードは、最後の文字が確定されたときや、ユーザーが選択または変換キーを押したときに終了し、それらのキーもまた、このモードでは通常の機能を有している。

【0036】本発明のさらに別の観点では、入力キー順序を明瞭にするために使用される単語と語句の基準データベースがツリーデータ構造を使用して語彙モジュールに好ましくは記憶される。特定のキーストローク順序に対応する単語は命令の形態でツリー構造に記憶されたデータから構成される。命令は直接先行するキーストローク順序（即ち最後のキーストロークを除く特定のキーストローク順序）に関する単語のセットまたは単語ステムを変更し、それによって現在のキーストロークが追加されているキーストローク順序に関する新しい単語セットおよび単語ステムを生成する。この方法による単語の構成は、ツリー構造の上部で単語ステムを構築する命令が一度だけ記憶され単語ステムから構成される全ての単語により共有されるので、語彙モジュールの記憶スペースを減少する。記憶されたオブジェクト、即ち例えば単語および単語語幹（ステム）を突き止めるための検索が必要とされないの、ツリー構造は処理の必要性を著しく減少する。ツリーデータ構造に記憶されたオブジェクトは、最初にユーザーに表示されるオブジェクトを示す頻度またはその他のランキング情報を含んでもよく、さらに処理の必要性を減少する。さらに、データベースがキーストローク順序に関連するオブジェクトを検索するた

めに使用されるとき、付加的な処理負担を起こすことなく、このツリーデータ構造はデータベースに必要な総寸法をさらに圧縮する特別なアルゴリズムを使用して変更されてもよい。

【0037】本発明のさらに別の観点では、ツリーデータ構造は2つのタイプの命令を含んでいる。第1の命令は単語および語句の発音に対応する仮名の順序からなる語彙モジュールに記憶された単語および語句の読み方を生成する。これは、各読み方に対応するのは第2の命令のリストであり、各読み方に関する見出し語を生成する。各読み方は直接先行するキーストローク順序に関して読み方のうちの1つを変更する第1の命令によって生成される。同様に、各見出し語は、第2の命令が関連される第1の命令によって変更された読み方に関して見出し語のうちの1つを変更する第2の命令によって生成される。

【0038】好ましい実施形態におけるキーの内部的な論理表示は実際のキーのラベルにより表された物理的配列を反映する必要はない。例えば、日本語の語彙モジュールを表すように構成されたデータベースでは、4つの付加的な文字（い、う、え、お）は1つの文字「あ」とだけラベルを付けられたキーと関連されてもよい。同様に、濁点および半濁点（゛と゜）等の特別な区分発音符号を有する文字も1つのキーと関連されることができ。例えば文字（き、く、け、こ、が、ぎ、ぐ、げ、ご）も1つの文字「か」とだけラベルを付けられたキーと関連されてもよい。これによってユーザーは区分発音符号を有する文字を含んだ単語を容易に呼出し、タイプすることができ、関連する標点文字の論理的に関連する物理的なキーを単に付勢することによって1文字当たり1回のみのキー付勢を行う。

【0039】さらに、本発明の別の観点では、任意の特定の関連する読みについてデータベース構造に各漢字文字を一度だけ記憶することによって、より大きなデータベース圧縮の効率が実現される。一般的に、データベースは、同じ読みを有する同じ漢字（例えば「動物」（どうぶつ）と「植物」（しょくぶつ）の漢字「物」（「ぶつ」と読む））の複数の異なった例を含んでいる。好ましい1実施形態では、ツリー構造の根の部分から直接開始して、漢字のコードの全ての特定を伴って所定の漢字に関する各読みがデータベースに含まれる。データベース（ツリー構造の根の部分から直接開始しない）中で同一読みを有するその他の漢字は間接的な基準により限定され、これはツリー構造の根の部分から直接開始した読みに関する見出し語のリストで十分に特定された漢字の相対的な位置を特定する。

【0040】上述のツリーデータ構造は、膨大な量の語のリストの変換やアクセスのために非常に効率的であるが、本発明で用いられている語彙モジュールは、他の構造を有することができ、この技術において、通常の技術

の内の一つを明らかにするであろう。特に、1つの実施の形態では、認識された一連のデータにユーザーのために語を追加していくメカニズムを持っている。用語の大部分を持っているデータベースは、しばしば読み出し専用メモリに格納されているので、使用しているときに変更はできない。その代わりとして、新しい語は、独立したユーザーデータベースに格納されており、このユーザーデータベースは、単純な読み方と関連づけられた見出し語のリストとして、もっとも効率的に格納されている。

【0041】多数の文字のキーへの割当てと、選択キーを使用した単語の定義と、選択キーを使用した所望の読み方の選択および選択的にそれに後続して変換キーを使用した所望の見出し語の選択と、選択リストの第1の単語として最も共通して使用される単語または単語ステムの表示と、選択リスト中の多数の解釈を含有と、後続する単語の第1のキーストロークによる順序への選択された単語の自動的な付加と、重大な処理ペナルティを受けることなく明瞭さのために大きいデータベースを圧縮する能力と、区分発音符号のない文字に関連するキーをタイプすることによる区分発音符号のある文字を有する単語を生成する能力とを組合わせた効果は驚異的な結果を生み、日本語では、テキスト材料の表示全体で見られる単語の99%が非常に高い効率でこのシステムにおいてタイプされることができる。平均して、各可能な基本的な仮名（即ち、図35で示されているキーを含む50のキーと小さい「っ」、「ゃ」、「ゅ」、「ょ」）に対して1つのキーを含む全てのキーボード上でテキストを入力することと比較して、1単語当たり0.61のみの付加的なキーストロークが12のキーしか有しない本発明の小型化されたキーボードを使用するときに必要なとされる。ローマ字を使用して所望の単語を綴ることによって仮名が入力される一般的なキーボードと比較して、平均してシステムは実際により少数のキーストロークしか必要としない。単語が区分発音符号を有する文字を含んでいるとき、付加的なキーストロークの節約が実現できる。単語が使用頻度順序で示されるとき、所望の単語は最も多くは最初に示される単語であり、多くの場合、唯一の示される単語である。ユーザーは付加的なキーストロークを必要とせずに次の単語を入力し続けることができる。それ故、テキストの高速度の入力が少数のキーを有するキーボードを使用して実現される。

【0042】日本語は、膠着性の強い言語である。そのような形態素（morphemes）（変化形、冠詞、前置詞等）、インドヨーロッパ語のような独立した語として表されるべきであり、日本語では、語のための接尾辞として書かれる。その代わりに、あてはまるステムと接尾辞のリストから、動的に生成された語なしで、ユーザーが入力したいすべての語の範囲を満足するのは困難である。このデータを入力するために、語（語彙項目）のル

ートについての情報を伴い、本発明の別の実施の形態は、1つまたはそれ以上のベースツリーを供給し、また、このデータを入力するために、語（形態素）の接尾辞についての情報を伴い、また、本発明の別の実施の形態は、1つまたはそれ以上の接尾辞ツリーを供給する。前記ルートと接尾辞自体の生成に加えて、また、これらのツリーは、ルートと接尾辞の唯一の規則的な組み合わせを割り当てるための必要な情報を備えており、そして、生成された完全な語（ルート及び／または接尾辞を伴ったルート）を相応に命令するための必要な情報を備えている。ルートと接尾辞からの語の動的な生成のプロセスは、「自動活用」（“auto-inflection”）として文書内で参照される。

【0043】この自動活用（auto-inflection）とユーザーデータベースとを備えたシステムでは、本発明の別の観点から、ユーザー定義語能力について、それらの活用のすべてのセットを供給することによって、さらに詳しく説明される。ユーザー定義語の接尾辞分類は、もっともありそうだと推測される分類（一般名詞）、あるいは、ユーザーが用いている単純なメニューインターフェイス、または、他の知られた技術を用いた方法によって、指定することができる。

【0044】ここで説明している小型化されたキーボードの曖昧化をなくすシステムはシステムを構成するコンピュータまたはその他の装置のサイズを減少する。キー数の減少により装置はユーザーが一方の手で動作しながら、他方の手で保持するように構成されることが可能である。説明したシステムはセルラ電話、PDA、両方向ページャ、または正確で高速度のテキスト入力から利点を得られるその他の小型電子装置で特に便利である。システムは圧縮されたデータベースを使用するとき付加的な処理帯域幅を必要とせずに曖昧性のないキーストローク順序用の大きなデータベースを効果的に圧縮する。タッチスクリーンベースの装置または、限定されたディスプレイスクリーン区域を有する限定された数の機械的キーを有する装置に構成するとき、システムは効率と簡潔さとの両者を与えることができる。

【0045】

【発明の実施の形態】（フローチャートに示されたブロックの説明）発明の実施形態を説明する前に、添付した図面中のフローチャートのブロックについて説明する。

【0046】図7：

150…キー入力待ち

151…キー入力はモード選択キーか？

152…キー入力は選択キーか？

152A…キー入力は変換キーか？

152B…選択リストは空か？

153…キーパッドモードのフラグはセットされたか？

154…キー入力をキー入力シーケンスに付加する

156…語彙モジュールにおいて対応する読み方オブジ

エクトを識別する

158…オペレーションのシステムモードにしたがって異なる語彙モジュールからのオブジェクトリスト間の優先順位を決める

160…同一の語彙モジュールからのオブジェクトの優先順位を決める

163…選択リストは空か？

165…選択リストを表示し、第1の読み方を仮に選択し、選択リストおよび挿入ポイントに表示する

166…選択リストにおいて選択されたアイテムはあるか？ 10

167…仮に表示されたエントリをテキスト領域の先のキー入力シークエンスの解釈として受け入れ、通常のテキストに変換する

168…キーパッドモードとキー入力に基づいて明白な文字を出力する

169…キーパッドモードは自動的に変更されるか？

170…キーパッドモードのフラグを先の値にもどす

171…新しいモードに必要なように表示を更新する

172…キーパッドモードのフラグをセットする 20

173…キーパッドモードは変わったか？

174…選択リスト中の第1の読み方を選択されたアイテムとして印を付け、挿入ポイントにテキストを仮に表示する

175…キー入力を待つ

176…キー入力は選択キーか？

177…キー入力は変換キーか？

図8：

178…選択リストの次の読み方に進み、選択されたアイテムとして印を付ける 30

179…選択リスト中及び挿入ポイントに選択された読み方を仮に表示する

180…仮に表示されていたエントリをテキスト領域のキー入力シークエンス解釈として受け付ける

184…古いキー入力シークエンスをクリアする

190…現在の読み方に関連付けられた最初の見出し語を選択リスト中の選択されたアイテムとして印を付け、見出し語テキストを挿入ポイントに表示する

191…キー入力を待つ

192…キー入力は選択キーか？ 40

193…キー入力は変換キーか？

194…現在の読み方に関連付けられた次の見出し語に進み、選択されたアイテムとして印を付ける

195…選択された見出し語を選択リスト中及び挿入ポイントに仮に表示する

196…現在選択されている見出し語に関連付けられた読み方に戻り、選択されたアイテムとして印を付ける

図9：

200…最初のキー入力のための現在のキー入力を設定

201…再スペルモードスタート 50

202…キー入力待ち

204…キー入力はデジットキーか？

206…キー入力は、現在のキー入力の有効な第2のキー入力か？

208…キー入力は、選択キーか？

210…排除する語のフィルター選択リストは、2キー入力によって入力確定された文字と一致しない

212…キー入力は変換キーか？

214…現在のキー入力は最後か？

216…現在のキー入力を促進する

218…選択リストと仮選択を表示しなさい、そして選択リスト中と挿入点における最初の読み方を表示する

図22：

600…キー操作リストによって、語彙モジュールから取り出されたオブジェクトの検索(RETRIEVE)サブルーチン開始

602…新たなオブジェクトリストをクリア(CLEAR)する

604…ツリーのルートノードのためにポイントする

606…最初のキー入力を得る

608…サブルーチンを呼び出して、現在のキーで入力されたキーを処理する

610…すべてのキー入力が処理された？

612…次のキー入力を得る

614…新たなオブジェクトリストに戻る

図23：

800…キー入力のリストから、語彙モジュール(自動活用の実施)の中より目的の語を検索する、サブルーチンスタート

801…キー入力の長さと等しくNを設定し、1と等しくMを設定する 30

802…第M番目のキー入力において、入力の最後と一致するすべての完全なルートワードを確認する

804…残りのN-Mキー入力と一致するすべての接尾辞を確認する

806…同じ分類の特定されたものを伴った接尾辞とルートワードのすべての組み合わせを作る

808…完全な語をこのキー入力のための有効なリストの語に加える

810…M=Nかどうかの特定を行う 40

812…1づつMを増加させる

814…新しい目的の語のリストに戻る

図24：

830…自動活用オブジェクトの頻度を計算、スタート

832…オブジェクトが、ベースツリーの中で完全であるか？

834…オブジェクトが非接尾辞としてつけないように表すことができるルートワードであるか？

840…接尾辞ツリーからの接尾辞の相対的な頻度とベースツリーからのルートの頻度の生成に戻る 50

8 4 2…ベースツリーからのルートの頻度に戻る
 8 4 4…この分類の非接尾辞としてつけない語のための相対的な頻度の不足とベースツリーからのルートの頻度の生成に戻る

図 2 5 :

8 5 0…選択リストにおける区分的に活用された語 (Sorting dynamically inflected words) のためのアルゴリズムを比較する、スタート

8 6 0…読み方は同じ頻度であるか?

8 6 2…一番高い頻度の読み方は、最初に持ってくる

8 6 4…異なる長さのルートワードか?

8 6 6…さらに長いルートワードを伴った読み方を最初に持ってくる

8 7 0…ルートワードは同じ語か?

8 7 2…接尾辞ツリー選択リスト中での最初の接尾辞に一致した読み方を一番最初に持ってくる

8 7 6…ベースツリー選択リスト中での最初のルートワードに一致した読み方を一番最初に持ってくる

図 2 6 :

8 7 8…変換リストにおける区分的に活用された語 (Sorting dynamically inflected words) のためのアルゴリズムを比較する、スタート

8 8 0…見出し語は同じ頻度であるか?

8 8 2…一番高い頻度の見出し語は、最初に持ってくる

8 8 4…関連した第 1 の命令は異なっているか?

8 8 6…第 1 の命令と関連した見出し語は、一番最初に持ってくる、図 2 5 と一致したものは一番最初にもってくる

8 9 0…同じ命令からの 2 つのルート見出し語か?

8 9 2…接尾辞に一致した見出し語は、接尾辞ツリー変換リスト中で最初であり、一番最初に持ってくる

8 9 6…ルートワードに一致した見出し語は、ベースツリー変換リスト中で最初であり、一番最初に持ってくる

図 2 7 :

6 2 0…キー入力を処理するサブルーチンスタート

6 2 1…現在のノードの有効キービットフィールドを読む

6 2 2…キー入力是有効か?

6 2 4…古いオブジェクトリストに戻る

6 2 6…古いオブジェクトリストのために新規なオブジェクトリストをコピーする

6 2 8…最初の有効キーの命令を呼び出す

6 3 0…NEW_INDEX=1

6 3 2…参照された、古いオブジェクトリストを呼び出すための命令のOBJECT-LIST-INDEXフィールドを用いる

6 3 4…命令LOGICAL-SYMBOL-INDEXフィールドを用い、呼び出されたオブジェクトに入る、現在のキーからの記号(SYMBOL)を追加する

6 3 6…NEWS-INDEXの位置にある新たなオブジェクトリストに入る変化(MODIFIED)されたオブジェクトを蓄積する

6 3 8…STOP-FLAGの命令はなされた?

6 4 0…新しい有効な命令を呼び出す

6 4 2…NEW-INDEXを増加させる

6 4 4…子ノードのためにポインターを追従(FOLLOW)させる

6 4 6…新たなオブジェクトリストに戻る

図 2 8 :

6 5 0…開始

6 5 2…それらの関連づけられたキーのための追加の不明確な記号を指定する

6 5 4…語彙リストにおける記号の使用頻度をカウントする

6 5 6…頻度の減少によるそれらの論理的キーでの記号のオーダー(ORDER)

6 5 8…語彙リストからのツリーを構築(BUILD)する

6 6 0…ツリーフォールディング(FOLDING)サブルーチンに入る

6 6 2…最も使用されたノードのために、すべての使用しない子ノードポインターをポイントする。

【0 0 4 7】 6 6 4…ホフマンエンコーディングを用い、最小の長さの命令とアドレスのビットパターンを指定する。

【0 0 4 8】 6 6 6…ファイルにデータを書き込む

6 6 8…終了

図 2 9 :

6 7 0…ツリーフォールディング(FOLDING)サブルーチンを開始

6 7 2…NODE-Aをルートノードの後の最初のノードに示す(POINT)。

【0 0 4 9】 6 7 4…サブルーチンツリーは、命令とSUB-NODESにおいて、NODE-Aに関連づけられた最高の重複度を有するノードを検索する。

【0 0 5 0】 6 7 6…目的(DESTINATION)が見つかった?

6 7 8…リコーシブリー(RECURSIVELY)にNODE-AとそのSUB-NODESを目的に合併させる

6 8 0…NODE-Aポイントは最後のノードか?

6 8 2…NODE-Aを次のツリーのノードに示す

6 8 4…戻る

図 3 0 :

6 9 0…最大重複度のツリー検索サブルーチン開始 趣旨(ARGUMENT): NODE-A

6 9 2…MAX-SAVINGS=0

6 9 4…NODE-Bをルートノードに示す

6 9 6…NODE-AとNODE-Bの重複度計算のサブルーチン

6 9 8…MAX-SAVINGSより重複度が大きい?

7 0 0…BEST-NODE=NODE-B, MAX-SAVINGS=REDUNDANCY

7 0 2…NODE-Bが最後のノードに示された?

7 0 4…NODE-Bを最後の次のツリーノードに示す

7 0 6…MAX-SAVINGとBEST-NODEに戻る

図 31 :

7 1 0...2つのノードの重複度計算: NODE-A、NODE-Bの
サブルーチン開始

7 1 2...SAVED_INSTRUCTIONS=0

7 1 4...LET KEY_INDEX=1

7 1 6...リスト-Aの中に入るNODE-Aの(KEY_INDEX)キ
ーからの入力命令

7 1 8...リスト-Bの中に入るNODE-Bの(KEY_INDEX)キ
ーからの入力命令

7 2 0...両方のリストは空?

7 2 2...各リストからの命令を得る

7 2 4...STOP-FLAGSを無視し、命令を行うことに同意す
る?

7 2 6...失敗(failure)に戻る

7 2 8...SAVED_INSTRUCTIONSを増加する

7 3 0...最後のキーで行われた?

7 3 2...KEY_INDEXを増加する

7 3 4...LET KEY_INDEX=1

7 3 6...NODE-AとNODE-Bの(KEY_INDEX)キーから、ポイ
ンターを得る

7 3 8...両方のポインターが空?

7 4 0...子ノード重複度計算のためにリコース(RECURS
E)する

7 4 2...比較が失敗したか?

7 4 4...失敗(failure)に戻る

7 4 6...SAVED_INSTRUCTIONを蓄積する

7 4 8...最後のキーが行われた?

7 5 0...SAVINGに戻る

7 5 2...KEY_INDEXを増加する

(システム構成及び基本的な動作)図1を参照しながら、本発明による1つのキーに2つ以上の文字を割り当てて小型化した曖昧性を解消するキーボード装置であって、ディスプレイ53を有する携帯型電話52に組み込まれたものを説明する。前記携帯型電話52は、一般的な電話のボタンによって実行される小型化されたキーボード54を備えている。この出願の目的に関して、“キーボード”という用語は、入力キーの範囲が定められているタッチスクリーン、個別の機械式のキー、膜型キー等を含む全ての入力装置を含むように広義に定義されている。キーボード54での各々のキー上の好ましいかな配列が、図1に示され、日本の電話機に関する事実上の基準に対応する。

【0051】そのキーボード54は標準的なクワータイ(QWERTY)キーボードにくらべて、あるいは図35に示してある日本語の基本的な文字表にある「かな」のそれぞれに一つのキーが割り当てられている、少なくとも46個のキーを持つキーボードにくらべて、データ入力キーの数が少ないことに着目されたい。更に具体的には、この実施形態で示される好ましいキーボードは、選択キー60と変換キー62と共に、3列4行に配列さ

れた、1から0までの番号が割り当てられている10個のデータキーを備えている。また選択的に、キーボードは、1つ前のキー入力を消すためのClearキー64や、あいまいでない(unambiguous)文字、数字、記号の入力のためのモードを入力するためのモード(mode)キー67や、そして前に入力したかなに濁点か半濁点を加えるための発音区別符(Diacritic)キー68を備えても良い。

【0052】データは、小型化されたキーボード54でのキー入力を介して曖昧さを解消するシステム(disambiguation system)に入力される。第一の望ましい実施形態では、ユーザーがキーボードを用いて連続したキー入力を行うと、テキストが電話のディスプレイ53上に表示される。図1、及び図2に表示される携帯電話のような、限られたディスプレイスペースしか有さない装置においてこのシステムが実行される場合、現在、選択されている、または、最も対象となりそうな言葉のみが、生成中のテキストにおける挿入ポイント88に表示される。所望の言葉を入力するためにキーが連続して押されると、一連の入力に対応する最も有り得る言葉は、ある特徴的な形式で表示される。図1に示されるこの望ましい実施の形態において、現在入力中の言葉は、点線による下線を伴って表示される。以下に更に詳細に説明されるように、選択キー60、または、変換キー62が押された後、点線の下線は実線の下線に変わる。

【0053】図5及び図6に示される第2の望ましい実施の形態では、情報をユーザーに表示するために2つの領域がディスプレイ53上で定義される。テキスト領域66は、上記第1の実施形態に関して記載したように、ユーザーによって入力されたテキストを表示し、テキストの入力及び編集用のバッファとしての役目を果たす。図5、及び図6に示されるように、通常テキスト領域66の下方の位置する選択リスト領域77は、ユーザーがキー入力した一連のキー入力に対応する語や他の解釈のリストを表示する。選択リスト領域は、その入力された一連のキー入力の最も頻度の高い解釈と、他のより頻度の低い別の解釈を頻度にしたがって降順で同時に表示することによって、ユーザーが入力されたキーからの入力の曖昧性を解消することを支援する。

【0054】縮小されたキーボードを用いた曖昧性解消システムのソフトウェアのブロックダイアグラム(block diagram)を、図4に示す。キーボード54とディスプレイ53とは、適切なインターフェイス回路を介してプロセッサ100に接続される。選択的に、スピーカ102が、プロセッサに接続される。プロセッサ100は、キーボードからの入力を受け、ディスプレイ、及び、スピーカへの出力のすべてを管理する。プロセッサ100はメモリ104に接続される。このメモリは、ランダムアクセスメモリ(RAM)のような一時記憶媒体と、読み専用メモリ(ROM)、フロッピー(登録商標)ディ

スク、ハードディスク、または、CD-ROMのような固定記憶媒体の組み合わせを含む。メモリ 104 は、システム・オペレーションを管理するための全てのソフトウェア・ルーチンを含む。好ましくは、このメモリは、オペレーティング・システム 106 と、曖昧性解消ソフトウェア 108 と、以下により詳細に論じられる語彙モジュール 110 を含む。選択的に、このメモリは 1 つ以上のアプリケーション・プログラム 112、114 を含んでいても良い。このアプリケーション・プログラムの例としては、語・プロセッサ、ソフトウェア辞書、及び、外国語翻訳ソフトがある。音声合成ソフトもまたアプリケーション・プログラムとして搭載されて良く、これにより、縮小したキーボードを用いた曖昧性解消システムが通信補助手段として機能することも可能となる。

【0055】図 1 に戻って、曖昧性を解消する小型化したキーボードによって、ユーザーは片手のみでテキストや他のデータを迅速に入力することが可能になる。データは小型化したキーボード 54 を用いて入力される。1 から 0 までの各々のデータキーは、文字、数、及び、他の記号によってキー上に表示された、複数の意味を持っている。(ここに開示されたものについては、データキー上に表示された数字と文字によって、各々のデータキーが識別される。例えば、「3 サ」によって右上のデータキーが識別される。) 個々のキーが複数の意味を持っているので、連続したキー入力はその意味が曖昧である。ユーザーがデータを入力すると、キー入力の様々な解釈がディスプレイ上の多重領域に表示され、ユーザーが曖昧性を解決することを支援する。大きいスクリーンを持つ装置では、入力済のキー入力の考えられ得る解釈の選択リストが、選択リスト領域においてユーザーに表示される。選択リストにおける第 1 のエントリ (entry) は、初期設定の解釈として選択され、そして、テキスト領域 66 の挿入ポイント 88 において表示される。この好ましい実施形態では、このエントリは、挿入ポイント 88 (そして大きいスクリーンを持つ装置では選択リスト領域内に) でその下に引かれた点線下線を伴って表示される。この形式は、このオブジェクトが現在の選択リストの最も頻度の高いオブジェクトであることによって、暗黙のうちに選択されることを意味する。表示装置が図 5 に示されるような選択リスト領域を含むならば、この形式はまた、挿入ポイント 88 におけるオブジェクトと選択リスト領域 77 に表示された、そのオブジェクトとを視覚的に関連付ける。図 1 においては、選択リストは表示されず、初期設定のオブジェクト (選択キーが活性化される前に選択リストにおいて最初に表示されるであろうオブジェクト) のみが、あるいは、あるオブジェクトが明確に選択されている場合にはその選択されているオブジェクトのみが挿入ポイント 88 に表示される。

【0056】入力されたキー入力の考えられ得る解釈の

選択リストは、いろいろな方法によって順序づけされてよい。通常モードのオペレーションにおいて、キー入力は、所望の語に対応する読み方をつづる仮名のエントリとして初めは解釈される(以降「語解釈」)。例えば、図 5 に示すように、「カ」「カ」「カ」という連続したキー入力ユーザーによって入力されている。キーが入力されると同時に、連続キー入力に合致した読み方を探し出すように語彙モジュールの参照が行われる。読み方のリストの最初に最も一般に使われる読み方を挙げるように、読み方は使用頻度に基づいて語彙モジュール内においてソートされている。先の連続キー入力の例をとってみると、読み方「かかく」、「きかく」、及び、「きこく」がその連続キー入力に対応する 3 つの最も有望な読み方として語彙モジュールより識別される。選択リスト中の 8 つの識別された読み方のうち、「かかく」が最も頻繁に使われるものである。従って、それは、初期設定の解釈として扱われ、挿入ポイント 88 にひらがなテキストとして仮に配置される。図 1 において示されるように、選択キー 60 が押下される前には、この第一の読み方は初期設定の解釈として扱われ、特有の形式を用いて挿入ポイント 88 に配置される。この形式は、新しいキー入力を開始するのではなく、データキーの内の一つによる次のキー入力が現在のキー順序に付加されることを示す。例えば、図 1 に示すように、この特有の形式は、点線下線を伴うひらがなテキストとして読み方を表示することからなる。他の考えられ得る読み方のリストはメモリの中に保存され、それらの相対使用頻度に従ってソートされる。

【0057】この好ましい実施形態においては、所望の読み方に対応する連続キー入力のエントリに続いて、ユーザーは選択キー 60 を単に押下する。挿入ポイント 88 で表示された初期値の読み方「かかく」の下点線下線は実線下線に置き換わる。表示された初期値の読み方が所望の読み方ではない場合、所望の読み方が現れるまで、選択キー 60 を繰り返して押す。1 つの好ましい実施の形態としては、メモリ内のキー入力に合致した読み方が、選択キー 60 の活性化の繰り返しによってすべて表示された後は、そのキー入力は数字として解釈され、それぞれのキー入力は各々のキー上に表記してある数字を生成する。このことによって、数が個別の数値モードなしで入力されることを可能にし、そしてまた、読み方解釈の選択リストの終りを簡単に認識できる表示としての役目を果たす。選択キー 60 をさらに押すと、選択リストの最初の読み方に戻るようになる。

【0058】いったん、所望の読み方が表示され、所望の「見出し語」が、実際ひらがな既に表示されている「読み方」と同じであるならば、ユーザーは、入力すべき次の読み方の最初のかなと対応するデータキーの押下を続行する。一方、所望の「見出し語」が漢字、漢字とひらがな、カタカナ、またはそれらの組み合わせから成

るならば、ユーザーは変換キー 6 2 を押下する。これによって、表示された読み方は語彙モジュールにおけるその読み方と関連している最も頻繁に発生している見出し語に置き換えられる。変換キー 6 2 を繰り返し押下すると、表示される見出し語は頻度の高いものから低いものへと置き換えられていく。また、好ましい実施形態において、変換キー 6 2 を繰り返し活性化させることによって、選択された「読み方」に関連付けられたメモリ中の「見出し語」が全て表示されると、選択された「読み方」はカタカナ（自動活用（auto-inflection）による実施の形態では、ただ語のルート部分でカタカナ中に表されている）で表示される。これによって、別個のモードを用いずにカタカナを生成することが可能となり、また、「見出し語」解釈の選択リストの最後であることを容易に認識できる表示としての役割を果たす。他の好ましい実施の形態においては、キー入力によって関連づけられたデフォルトの読み方をユーザーが関連づけを希望する場合には、変換キー 6 2 を即座に押すことによって、所望の見出し語が初めに選択キー 6 0 を押すことなく関連づけされる。

【0059】選択キー 6 0 か変換キー 6 2 のどちらか 1 つまたは両方を一回以上押下した後、何かデータキーを押下することによって、表示されている「読み方」や「見出し語」の特別な形式（この好ましい実施形態では実線下線）が除去され、新たなキーシークエンスの最初のキー入力、システムによって解釈されるようになる。先行するキー入力シークエンスの解釈を確認するために特別にキー入力する必要はない。

【0060】上述の好ましい実施態様において、選択キー 6 0 を押下すること現在のキーシークエンスに関連付けられたメモリ中の読み方を順方向に循環させる（頻度について降順）。別の実施形態においては、選択キー 6 0 を所定の時間しきい値を超えて押下しつづけることによって、メモリ中の読み方を頻度について昇順に循環させる。すなわち、上述のような、メモリ中の一連の関連付けられた読み方の終りに数字解釈が含まれている場合には、選択キーの通常押下に先立って、選択キーが押下しつづけることによって、直ちに数値解釈に戻る。選択キー 6 0 を押下しつづけることを繰り返すと、関連付けられた読み方を昇順で逆方向に循環させる。同様に、変換キー 6 2 を押下しつづけることによって、現在選択された読み方に関連付けられた見出し語を昇順で逆方向に循環させる。同じく、変換キー 6 2 の通常押下に先立つ最初の変換キー押下継続操作によって、直ちにカタカナ解釈に逆進循環する。

【0061】図 1 についてさらに言及すれば、他の好ましい実施形態においては、データキーの入力があると、入力されたデータキーを削除するためにクリアキー 6 4 を押しても良い。現在のキー入力シークエンスの全てのデータキーがその結果削除されている場合、クリアキー

6 4 を押すと、挿入ポイント 8 8 の左側にある、テキスト表示 5 3 上の文字が削除され、現在の選択リストがからの場合は、標準のテキストカーソルが表示される。選択キー 6 0 か変換キー 6 2 のどちらか 1 つまたは両方を一回以上押下した後、クリアキー 6 4 を押すことにより、挿入ポイント 8 8 にある現在選択された原文解釈を現在のキー入力の初期値の「読み方」解釈に置換えるが、連続したキー入力からのデータキーのいずれも削除するものではない。言い換えれば、選択キー 6 0 および / または変換キー 6 2 を何回か押下した後にクリアキー 6 4 を押すと、選択キー 6 0 と変換キー 6 2 の全ての活性化は取り消しとなり、選択キー 6 0 か変換キー 6 0 の初めのキー入力直前の状態にシステムを戻す。別の実施の形態においては、変換キー 6 2 が一回以上押下された後、選択キー 6 0 が押下されると、挿入ポイント 8 8 において現在選択されている「見出し語」をその「見出し語」に関連付けられた「読み方」に置換える。さらに選択キー 6 0 を押下すると、現在のキー入力に関連付けられたメモリ中の他の「読み方」についてその位置から順方向（頻度について降順）に循環しつづける。

【0062】また、他の実施形態においては、（特別な記号モードでの入力や、単一の特定の文字に明確に関連づけられたキーの押下のような）曖昧でない文字を生成する他の手段の活性化は、現在のキー入力を終了させる働きをする。その結果として、挿入ポイント 8 8 に表示された「読み方」や「見出し語」の特別な形式（前記の実施の形態においては点線下線や実線下線）は削除され、その特定の曖昧でなくなった文字が新しい挿入ポイント 8 8 における出力された語に付け加えられる。

【0063】選択された見出し語や読み方をテキスト領域の挿入ポイント 8 8 に仮に配置することによって、選択リストを参照することなく、ユーザーはテキスト領域に注意を向けていることが可能となる。ユーザーの選択により、選択キー 6 0 （または変換キー 6 2 ）の最初の入力を受け付けたときに、挿入ポイント 8 8 に仮に配置された「読み方」（または「見出し語」）を（垂直方向あるいは水平方向に）拡張し、現在の選択リストの複製を表示しても良い。ユーザーは、この選択リストの複製に表示される最大数の語を選択できる。あるいは、ユーザーは、たとえ選択キーの最初の活性化の前であっても常に選択リストを挿入ポイント 8 8 において表示させることを選択しても良い。この曖昧性を解消するシステムは、（曖昧なデータキーの活性化または明確かつ曖昧でない文字の生成によって知らされる）次の語の開始を選択された入力、所望の入力であるとの肯定として解釈する。従って、選択された語がユーザーの選択として挿入ポイント 8 8 に残り、下線は完全になくなり、その語は特別な形式のない通常のテキストとして再表示される。

【0064】テキスト入力の大部分において、連続したキー入力は、かなの読み方の形式として、ユーザーによ

って意図される。しかしながら、各キーに関連付けられた複数の文字と記号によって、各々のキー入力や連続したキー入力に複数の解釈を与えることができることが理解されよう。この縮小されたキーボードによる曖昧性解消システムにおいては、連続したキー入力が解釈されユーザーに語のリストとして表示されると同時に、様々な違った解釈が自動的に判断されて、ユーザーに表示される。

【0065】例えば、連続したキー入力は、ユーザーが入力していると考えられる一連の仮名に対応する語幹の観点から解釈される（以降「語幹解釈」）。語解釈と異なり、語幹は不完全な語である。直前のキー入力の考えられる解釈を表示することによって、語幹を用いることにより、ユーザーは正しいキー入力をおこなったかどうかを容易に確認でき、また、語の入力途中でユーザーの注意が反れた場合に入力を再開することもできる。長い語または句の部分的な入力に対応するが完全な語や句には対応しない連続したキー入力があるとする。そのような場合において、ユーザーに提供することができる最も有用なフィードバックは、その点までに入力された語の語幹に対応する仮名を表示することである。図5の例においては、「カ、カ、カ」という連続したキー入力は、（単語「かけきん」に先行する）語幹「かけき」を形成するものとして解釈することができる。従って、この語幹解釈は選択リスト中のエントリとして提供される。望ましくは、データキーによる次のキー入力によって各語幹から生成され得る、全ての考えられ得る語の組み合わせの合成頻度にしたがって、語幹解釈がソートされる。その結果、語幹解釈のいくつかは表示されないかもしれない。語幹解釈を選択リストにおいてリストするとき、選択リストに表示された語と重複する語幹解釈は省かれる。しかし、その語幹が省かれた場合には、その省略された語幹に対応する語は、その語を語幹として有する、より長い語があることを示す記号によって印を付けられても良い。所望の語のエントリにつながる正しいキー入力がなされたことを確認することによって、語幹解釈はユーザーにフィードバックを行う。

【0066】図7、8は、選択リストを処理し、かつユーザーが曖昧なキー入力シークエンスを曖昧でなくすることを支援するために、挿入ポイント88において何が表示されるかを決定する曖昧性解消ソフトウェアの主要なルーチンを示すフローチャートである。ブロック150においては、システムはキーボード54からのキー入力を待っている。キー入力を受けると、判定ブロック151において、受けたキー入力モード選択キーである

かどうかを決定するテストが行われる。もしそうであれば、ブロック172において、フラッグを現在のシステムモードに設定する。

【0067】判定ブロック173において、システムモードが変化しているかを判断するテストが行われる。もしそうであるならば、ブロック171において、現在のシステムモードを反映するのに必要とされるように、ディスプレイが更新される。

【0068】ブロック151において、キー入力は、モード選択キーではないと決定するならば、ブロック152に移行し、ここでのテストは、受け取られたキー入力を選択キーであるかどうかを決定するために行われる。キー入力を選択キーでない場合は、ブロック152Aに移り、ここでのテストは、受けたキー入力に変換キーであるかどうかを決定するために行われる。もし受けたキー入力に変換キーでない場合は、ブロック153に移行する。ここでのテストは、システムが、記号モードのような特別な文字モードであるかどうかを決定するために行われる。もしそうであれば、ブロック166に移行し、ここでのテストは、選択リスト内に一時的な選択アイテムが存在するかどうかを決定するために行われる。もしそうであるならば、ブロック167に移行し、前記アイテムは容認され、通常のテキストとして出力される。そして、ブロック168に移行し、キー入力と一致した文字が、テキスト領域に出力される。そうすると、判定ブロック169に移行し、ここでのテストは、記号モードの場合と同様に、システムモードを自動的に変化させるべきかどうかを決定するために行われる。もしそうであるならば、ブロック170に移行し、システムモードは以前のアクティブ・モードに戻り、そうでなければ、実行はブロック150に戻る。もし、ブロック153において明白文字モードがアクティブでなければ、ブロック154において、記憶されているキー入力シークエンスにキー入力追加される。ブロック156では、キー入力シークエンスに対応するオブジェクトが、システムにある語彙モジュールから識別される。前記語彙モジュールは、キー入力シークエンスと関連づけられたオブジェクトのライブラリである。オブジェクトは、入力されたキー操作に基づいて取り出された、蓄積データの一部である。例えば、語彙モジュール内のオブジェクトは、数字、文字、語、語幹、句、またはシステム機能とマクロである。これらのオブジェクトのそれぞれは、以下の表に簡潔に記載される。

【0069】

【表1】

オブジェクト	対応するデータ
数字	数字、各数字は単独のキー入力に対応、例 2桁のシーケンス「42」。
文字	キー入力の組み合わせに対応する文字あるいは文字の連続、例 3文字シーケンス「きかく」、キー入力の各組み合わせは、曖昧でない文字を入力するための2回の押下による特定方法を用いて曖昧でなくするために使用される。
語	1回または複数回のキー入力に対応する読み方または見出し語。例 4文字からなる語「かけきん」。
語幹	ある語をなす仮名の長い連続の有効な部分を表す仮名の連続。例 語「かけきん」の語幹「かけき」。
句	1回または複数回のキー入力に対応するユーザまたはシステムに定義された句。例 「おねがいします」。
システム・マクロ	システムまたはユーザに定義された機能を記述する語または対応するコード。例 現在のテキスト領域をクリアするための「<clear>」。記述的な語に加えて、語彙モジュールにおいては、システム・マクロ・オブジェクトは特定された機能を実行するために必要な、実行可能なコードに関連付けられる。

【0070】前記のように、好ましい語彙オブジェクトが検討される一方、他のオブジェクトも考慮されてよい。例えば、グラフィック・オブジェクトは、蓄積されているグラフィックイメージに関連づけられており、また、スピーチ・オブジェクトは、蓄積されているスピーチの一部分に関連づけられている。スペル・オブジェクトもまた、一般的なスペルミスやタイプミスを正しいスペルに関連づけるものとして想定されて良い。処理を簡単化するため、各々の語彙モジュールは、好ましくは類似のオブジェクトを備えている。類似のオブジェクトは、認識されるが、様々なオブジェクトは、語彙モジュール内でミックスされても良い。

【0071】次に図7に戻り、ブロック156では、受け取ったキー入力シーケンスに対応するこれらのオブジェクトは各語彙モジュールにおいて識別される。ブロック158からブロック165までにおいては、語彙モジュールにおいてキー入力を調べることによって発見されたオブジェクトは、オブジェクトをユーザーに表示する際の順番を決定するために優先順位づけをされる。選択リストにおいて表示されたオブジェクトの順序を決定するために、優先順位は、各語彙モジュールの間で、更に、各語彙モジュールからの返されたオブジェクトの間で確立される。

【0072】様々な語彙モジュールから確認されたオブジェクトリストを優先順位づけするために、ブロック158では、縮小されたキーボードによる曖昧性解消シ

ステムのオペレーションモードが調べられる。上でのべたように、オペレーションの通常モードでは、語解釈（読み方及び見出し語）は、最初に選択リストにおいて表示される。したがって、語彙モジュールからのオブジェクトリストは、他の語彙モジュールからのオブジェクトリストより高い優先順位を割り当てられるであろう。逆に、曖昧性解消システムがオペレーションの数値モードにあるならば、数値解釈は、他の語彙モジュールより更に高い優先順位を割り当てられるであろう。曖昧性解消システムのモードは、従って語彙モジュールオブジェクトリスト間の優先順位を決定する。あるモードにおいてある語彙モジュールからのオブジェクトリストが完全に選択リストから省略されるかもしれないということが認識されるであろう。

【0073】語彙モジュールから生成されたオブジェクトリストは、シングルエントリーのみ含むかもしれない、または、それらは、複数エントリーを含むかもしれない。ブロック160では、オブジェクトリストが複数エントリーを含むならば、同一の語彙モジュールからのオブジェクトの間の優先順位付けは解決される。所定の語彙モジュールにおいて調べられた特別なキー入力シーケンスに合致するオブジェクトは、相互に関してそれらの相対的評価を決定する優先順位を同じく与えられる。上記で説明したように、望ましい標準の表示の順序は、よく使う代表的な選択肢の優先順位を減少することによるものである。各オブジェクトと関連していた優先順位デ

ータは、選択リスト中のオブジェクトを順番付けるために使用される。

【0074】語彙モジュールで調べられたオブジェクトの表示と関連づけられていた、多数の詳細事項は、適切なシステムメニューにアクセスすることによってユーザーによってプログラム可能である。例えば、ユーザーは、個々のオブジェクトのオーダ、または、選択リストの目的の分類を指定することができる。また、ユーザーは、語彙モジュールの間の、そして、各語彙モジュールから確認されたオブジェクトの間の優先順位を規定する優先順位レベルを同じく設定することもできる。

【0075】オブジェクト間の優先順位が決定された後で、ブロック165では、選択リストが、識別されたオブジェクトから組み立てられ、そして、ユーザーに提示される。ユーザーによって入力されたあいまいなキー入力順序の初期値の解釈として、図1、及び、図2示すように、選択リストにおける最初のエントリは、テキスト領域53における挿入ポイント88で仮に配置され、強調表示される。そして、曖昧性解消ソフトウェアのルーチンは、次にキー入力を待つために、ブロック150に戻る。

【0076】検出されたキー入力が選択キー60であるならば、「イエス」の選択肢は、ブロック152からブロック163に移る。ここでは、現在の選択リストが空であるかどうかテストされる。もし、そうであるならば、ブロック150に戻ることになる。もし、判定ブロック163において、選択リストが空である場合には、“ノー”の選択肢になりブロック174に進むことになる。ブロック174では、それが一時的にポストされた挿入ポイント88で表示されたデフォルトの読み方の下の点線の下線は、実線の下線に置き換えられる。ブロック175で、システムは、それからユーザーによって入力された次のキー入力を検出するのを待つ。キー入力を受け取り次第、ブロック176において、テストは、次のキー入力が選択キーであるかどうかを決定するために行われる。次のキー入力が選択キーであるならば、ブロック178では、システムは、選択リストの中を次の読み方に向かって進み、そして、それを現在の選択アイテムとして示すことになる。ブロック179では、現在の選択されたエントリは、実線下線を伴って挿入ポイントで一時的に表示される。ルーチンはそれからブロック175に戻り、ユーザーによって入力された次のキー入力を検出する。ブロック175からブロック179によって形成されたループによって、ユーザーは入力されたあいまいなキー入力シークエンスの様々な読み方解釈であって使用頻度の少ないものを選択することを可能にするということが認識されるであろう。このことで、ユーザーが選択キーを何回も押すことにより、使用の頻度が減少していくということである。

【0077】次のキー入力が選択キーではないならば、

決定ブロック177において、ここでのテストは、次のキー入力の変換キーであるかどうかを決定するために行われる。検出されたキー入力の変換キーであるならば、「イエス」の選択肢を通して、ブロック177からブロック190に進む。もし検出された打ち込みキーが変換キーならば、「イエス」の選択肢を選び、ブロック177から、最新の「読み方」に関連している最初の「見出し語」が選択項目としてマークされるとともに「見出し語」テキストが挿入ポイント88にて実線の下線によって一時的に表示されているブロック190に移行する。ブロック191では、システムは、ユーザーによる次のキー入力を待っている。ブロック192でのテストは、キー入力を受け取り次第、次のキー入力が選択キーであるかどうかを決定するために行われる。次のキー入力が選択キーであるならば、ブロック196では、現在選択された見出し語をの関連づけされている読み方に、現在選択されているアイテムを戻す。さらにそれを選択されたアイテムとして表示し、ブロック179に進む。ブロック192で次のキー入力が選択キーではないならば、ブロック193でのテストは、次のキー入力の変換キーであるかどうかを決定するために行われる。それが変換キーであるならば、ブロック194では、現在選択されているオブジェクトは、現在の読み方と関連している見出し語に進み、選択されたアイテムとして表示される。ブロック195では、現在選択された見出し語は、実線の下線による挿入ポイント88で一時的に表示される。また、システムは、ユーザーによって入力された次のキー入力を検出するためにブロック191に戻る。

【0078】ブロック177、及び、193で次のキー入力の変換キーではないならば、ルーチンは、ブロック180に移行し、ここで表示された一時的な入力、キー入力として選択されるか、テキスト領域に、ノーマルテキストとして表示される。ブロック184では、以前のキー入力は、システムメモリから消去され、選択キーが変換キーの後に、不確定なキーが入力されると、システムは、次の不確定な入力最初のところから入力を行うようになる。新しくキーが入力されたならば、ブロック154では、新しいキー入力変換を始める。最も高い使用頻度の読み方の候補がデフォルト選択として示されるので、ユーザーが選択キーを押す回数を最小にでき、効率よくテキストを入力することを可能にする。

【0079】(II. 曖昧でないテキストエントリシステム) 同様に本発明である、日本語入力のための小型化したキーボードは、ユーザーが望む入力要求の各々の所望のかなを指定することができる。日本語の五十音図には、108の音節がある(小さい“つ”、“や”、“ゆ”そして“よ”が、普通の大きさの“つ”、“や”、“ゆ”、“よ”とは違う文字として存在し、書かれたり発音されたりしている)。さらに、あまり使われない音節として、小さい“あ”、“い”、“う”、

“え”、“お”が存在する。これらは、本来はカタカナとしてだけ用いられる。これらのあまり使われない音節はまた、以下に説明するように、本発明のシステムから簡単に入力することができる。108の標準の音節のうちで、37は発音区別符、もしくは、他の71の音節のうちの1つの音節に「[˙]」か「[˚]」を1つを簡単に加えることによって入力できる。発音区別符なしのこれらの71の音節は、論理学上9のマトリックス、または、下で説明する10の行、及び、8～10の列に組織されていることができる。本発明のキーボード上のキーのほとんどには、かなの表示がついている。ある列のマトリックスによって関連していた子音を表すものと、また、マトリックスのある列によって関連していた母音を表す2番目のかなと共に分類される。

【0080】構成は論理的で、日本語のネイティブスピーカーにとって108の音節中の106の音節は直感的であり、残りの2つの音節（小さい“っ”と“ん”）は、単純で容易である。全ての音節は、2の別々のかな（例えば、KYA、KYU、及び、KYO）で表される口蓋音化された母音を含む、一対のキー入力によって作成される。いくつかの重要なキー入力において、現在用いられている一般的な入力方法によって、変換結果が表示される。このように、本発明のキーボードを用いることにより、日本語のネイティブスピーカーが容易に習得し得ることができ、キー入力にかかる時間を効率的に短縮することができる。

【0081】本発明の実施の形態において、日本語の音節表の71の音節は、図38で示された五十音表からなっている。図38において、最初の8つの列に現れる全ての69の音節を含むとき、一致するペアの最初のキー入力、出力されるために、音節の子音を決定し、第2のキー入力は、母音を決定する。残りの2つの音節（小さい“っ”、及び、“ん”）は、次に説明する例外的なケースである。図10で示されなかった残りの37個の音節は、図10の五十音表にある、一致する基本音節を生成し、その後、個別のキーを使う発音区別符を加えることによって表示される。図2に示した、画面領域が限界である携帯電話の概略図では、キーパッド（膜キーボード）に、日本語入力に関する本発明の実施の形態が組み込まれている。キーボードのテンキーである、図2の左上にふられたかなの121から130のそれぞれのキーは、図38の“Key1”に対応し、右下にふられたかなは、“Key2”に対応する。本発明の実施の形態においては、1つ目のキー入力が行で配列される左上のかなで入力し、2つ目のキー入力は、縦の列で配列される右下のかなで入力することで、図38での音節が入力される。図2に示すように、テンキー121～130の左上にある一般的な10個の列になっているかながふられたキーボードが表されている。最初の縦の5列のかなは、121～125のキーの右下を表している。小さな

“や”、“ゆ”、“よ”は、次の3つの126～128のキーに割り当てられている。最後に、小さな“っ”は、129のキーに、“ん”は130のキーに割り当てられている。発音区別符は、131のキーで発音区別符「[˙]」や「[˚]」を付加することで入力できる。母音を伴って、発音区別符を入力するとき（表4の縦の列“い”の後に、小さな“や”、“ゆ”、“よ”がつくときは、2つのかなで表される）は、発音区別符は、即座に2つのかなの後に加えられ表示される。

【0082】このように、一致する最初のキー操作が音節の子音を決定し、第2のキー操作は、母音を決定する。他の実施の形態においては、ディスプレイは、フィードバックをユーザーに行うために使われる。最初のキー操作を受け取り次第、第2のキー操作で確定され生成され得る様々な音節を表示する。数字又は他の入力キーに関連づけられた、各々のキーにかな等をふっていくことによって、キー入力に関連づけられる。代わりとして、音節は、一致するキーの配列と一致したときに、表示されることができる（変換候補番号がついて、もしくは、つかないで）。例えば、テキストディスプレイによる携帯電話のような装置の場合、音節は、1から9までの電話のキーの配列とのような、3行3列のマトリックスにおいて表示することができる。ディスプレイがこの方式を用いると、母音の音節の小さい“あ”、“い”、“う”、“え”、及び、“お”のようなめったに使われた音節を容易に入力することができる。例えば、図39に示すように、マトリックスを用いると、図39の上の列にある母音のみと一致する、最初のキー操作を受け取り次第、1～5のキーに関連づけられた普通の大きさの“あ”、“い”、“う”、“え”、“お”が表示され、小さい“あ”、“い”、“う”、“え”、“お”は、6～0のキーに関連づけられている。

【0083】図3はテキストディスプレイを持った携帯電話で、日本語小型化されたキーボード実施の形態である。テンキーである121～130のすべてのキーは、図39に基づき、キーの真ん中のところにながふられている。図3に示すように、121～130のキーには、10個の列のかなが表示されている。また、図3は、ディスプレイ表示が122のキー（“か”、“き”、“く”、“け”、“こ”に関連づけられている）によって、アクティブになった状態を示している。実施の形態においては、図39に表されている音節は、アクティブになっている列のキーにながふられたキーが操作されたときに表示される。ディスプレイには、変換されるかなとかな表変換のための2番目の入力された各々のキーとが合わさって表示される。図39における縦の列は、図3に示すような携帯電話のキー上に、数字の表示をつけて表されている。実施の形態については、母音を含む3音節（きや、きゅ、きょ）は、127～129に関連づけられており（キーの表示では、7、8、

9に割り当てられている)、これらは、行、1列で表示することができる。これによって、ユーザーが、これらの関連する音節をキーに関連づけして、入力しやすいようにすることができ、音節とキーの数字表示に関連づけるだけでなく、キーパッド上のディスプレイとキーで、音節の3行3列のマトリックスで、関連づけることができる。また前記のように、発音区別符は、131にキーを一回押すことによって、「[˙]」が加えられ、2回押すことによって、「[˚]」が加えられる。発音区別符が、母音を伴って加えられたとき(図3に示すように、キー127~129上に2番目のキーが関連づけられる)、発音区別符は、すぐに前記かなのすぐ後に加えられて表示される。

【0084】図5と図6に示すように、本発明の実施の形態では、携帯型コンピュータのタッチスクリーン上に、表示されているキーボードとして組み込んだ場合、変換候補を表示することができる。図5に示すように、キーボード150は、あらかじめ最初のキー入力を表示することができる。図6に示すように、キーボード170は、最初のキー入力として、“か”、“き”、“く”、“け”、“こ”を、図5のキー152のアクティブにしたキー入力の後に、表示することができる。

【0085】図6に示すように、キー171から179は、キーがアクティブになったときに、変換されるかなと共に表示される。キーや音節の配列は、図3に示すように、3行3列のマトリックスを利用したものであり、図39にあるように、キーに音節を関連づけることは、事前に音節を変換することができる。

【0086】別の実施の形態においては、図10に示すように、空のセルに関連づけられたキーの操作では、何も入力されず、ユーザーのオプション設定で、エラー音をならすこともできる。また、別の実施の形態においては、図2(小さい“っ”の表示があるのキー)に示すように、キー129を2回押すことによって、小さい“っ”が入力される。また、他の実施の形態においては、小さい“っ”はふられたキーは、ちいさな“っ”を入力する。同様に、別の実施の形態においても、“ん”とふられたキーは、“ん”を入力する。様々な変換は、音の変化は図38や4bを発明の目的から、はずれることなく、形成することができる。

【0087】例えば、小さい“っ”と“ん”は、例外的な場合であり、1回押すと小さい“っ”、2回押すと“ん”が生ずる分離キーでそれらを割りあてるような、広く様々な方法により指定される。さらに本発明の重要な点は、図38の最初の8行(または、図39の1-5行目と7-9行目)についての、音節の詳しい使い方である。

【0088】たとえ、図2に示した実施の形態におけるキーの行のラベルのかなおよび列のラベルのかなの割り当てが一番自然であれ、本発明の範囲から逸脱すること

なく、他の組み合わせにより、キーの行のラベルのかな及び列のラベルのかなを割り当てる他の具体例が可能であることは当業者にとり自明であろう。図10と図11は他の配列により行のラベルのかな及び列のラベルのかながキーに割り当てられた多くの可能な他の具体例のあるものを示す。

【0089】別の実施の形態では、本発明のシステムは図11に示すように9個のキーのみを有するキーボードを組み込むことができる。

10 【0090】図38の71個の別個の音節の68個もまた図40に示すように構成することができ、ここで行のラベル“や”に対応する行は削除される。かかる9個のキーシステムにおいては、3つの音節“や”、“ゆ”、“よ”は指定された1対のキー順序から生成され、ここで第1のキーは表の初めの行にあるセルに対応する行のラベルの“あ”である。小さな“つ”と“ん”は、最後の列のセルに対応するキー順序より生成され、これは互いに組合わされた図38の最後の2つの列を含む。

20 【0091】さらに、本発明の他の面においては、システムは、次のキー入力指定された1対の第1のキー入力のように処理される(出力される音節の母音の決定)か、もしくは指定された1対の第2のキー入力(出力される音節の子音の決定)のように処理されるかの命令をユーザーに提供する。例えば、この命令は、交互に点灯するラベル付けされた1対のLEDとして実施される。代わりに、2つのアイコンがディスプレイ上に交互に表示される。かかる命令の多くの他の代替する具体例が本発明の範囲を逸脱することなく可能であることも、当業者にとり自明である。

30 【0092】図2および図6に示すようなキーへの適切なラベルの配置の組み合わせの効果、命令された1対のキー入力による音節の生成、図3及び図6に示すような指定された1対の第1のキー入力に続く適切なフィードバックの提供、キーのラベルの編成と標準的な順序と日本語の音節文字表と間の自然な対応、および次のキー入力音節の子音を特定する行のラベルのかなまたは母音を特定する列のラベルのかなに対応するかどうかのユーザーに対する命令を含むことは、明確なテキスト入力に関し有効であり、日本語のネイティブスピーカーによる正確な入力と容易な理解および早い学習ができるという効果がある。テキストを早く入力するということは、少数のフルサイズのキーからなるキーボードを使うことによってそれ故に実現でき、使い方を習得したり実際に使用することが容易である。

40 【0093】111. 曖昧なおよび曖昧でないキー入力を組合わせるテキスト入力方法

また、本発明の他の面においては、音節の曖昧なもの(ambiguous)と曖昧でないもの(unambiguous)の両方を組合わせることでより効率的な入力方法を実現できる。あるの実施の形態においては、ユーザー選択で、入

力される語の第1の音節は上記2回のキー入力を使用して曖昧でなく特定される。口蓋音で発音される母音を有する音節の場合は、これらの第1の2回のキー入力は2つのひらがな文字（口蓋音で発音される母音を示す小さい“ゃ”、“ゅ”、“ょ”を含む）を特定する結果となることに注意されたい。残りの語や句の音節は各音節に対し1回のキー入力で曖昧に特定される。1または2音節の短い語の場合、この組み合わせ方法は入力時の変換確定作業を十分に軽減させ、その結果、ユーザーが入力したい読み方を見出すために試験を必要とする候補の数を減らすことができる。また他の代りの実施の形態では、語や句の第1の音節のみを曖昧でなく特定するために、この技術において既知の複数ストローク特定方法を使用することができ、残りの音節は曖昧に確定される。

【0094】さらにまた本発明の曖昧な特定方法を使用する別の実施の形態においては、2入力方法は語又は句（最初の部分だけよりはむしろ）のいずれかが望みの音節を特定するのに使用することができる。例えば、キーを押しある特定の時間押を過ぎることは、直接続くキー入力が望みの音節を曖昧でなく特定するための2入力の2番目のであることを示すことになる。これには、2つの利点が存在する。第1は、いずれかの音節（語や句の最初の音節のみよりもむしろ）が曖昧でなく決定されるということである。第2は、これが入力の効率を増加すると考えられる場合にのみ、ユーザーは音節を曖昧でなく特定することを選択することができるということである。全ての音節が曖昧に特定される場合、ある種の非共通の語および句に関し（そして特に短い語に関し）、最も共通に使用される語および句のデフォルト表示は、望みの読み方を選択するために選択キーの複数の操作を要求することができる。かかる場合、ユーザーは、同じ位置の同じ曖昧でなく特定された音節を共にする読み方の中から選択することのみを必要とするので、少なくとも1つの音節の曖昧でない特定は、十分に少ないキー入力の要求で望みの読み方を生成するという結果をもたらす。

【0095】また、別の実施の形態においては、再変換モードとして知られている、1つの語の完全なキー入力の後に、ユーザーに1つまたはそれ以上の入力された音節を明らかに特定させる。これはキー入力やメニュー表示させること、または他のいくつかの方法によって呼び出されうる。このとき、語は図9に示されるように曖昧さを解消される。まず、ブロック200において、現在のキー入力は、完全な語の最初のキー入力に設定される。ブロック204では、入力されたキーが、ディジットキー（図1のキー21～30）かどうか決定される。もしそうでなければ、選択キーもしくは変換キーはこのモードを終了し、このとき、標準音節入力モードで、アクティブになるということを決定ブロック208と212で示している。そうでない場合（ディジットキ

ーで会った場合）では、新しいディジットキーが押された時、2キー入力の第2のキーとして確定された（explicit）音節が入力される。ブロック206では、第2キー入力のために有効な第2キー入力かどうか決定される。もしそうでなければ、システムはエラー音（例えばビーブ音等）を発し、ブロック202へ戻り、次のキー入力の入力待ち状態となる。もし、キー入力が有効であれば、ディジットキーが押されるたびに、選択リストは、その位置と新しいキーの前入力キーによって、選択リストは、現在の位置でキャラクタを明確に特定される文字だけとするためにフィルターをかけられる（ブロック210）。もし、すべてのキーの曖昧さが解消されたときには、システムは自動的にユーザーが選択キーを押した状態に戻る（ブロック214）。そうでない場合は、現在のキーは自動的に次の位置に進められ、システムは選択リストを表示し、そして、挿入点に置いて、選択リスト中で最初の読み方の仮の選択と表示を行なう（ブロック216、218）。このとき、システムは新たなキーの待ち受け状態となる（ブロック202）。好ましい実施の形態において、入力された有効な第2のキー入力と、関連づけられた文字とのメニューは、システムが再変換モード状態にあるとき、ユーザーに表示される。

【0096】IV. 曖昧なキー入力に基づくテキスト入力を支持するデータベース構造

入力順序を明確にするために使用される語及び句の標準的なデータベースは、ツリーデータ構造を用いる語彙モジュールに蓄積される。特別のキー入力順序に対応する語は、ツリーデータ構造内において、語の組みおよびそれに付随する新しいキー入力を有するキー入力順序と結合された語幹（word stem）を変更する命令の形式で記憶されたデータから構成される。このように、順序についてのそれぞれの新たなキー入力が行われることによって、そのキー入力と結合された命令の組が語およびそれに付随する新しいキー入力を有するキー入力順序と結合された語幹の新しい組を創作するために使用される。このようにして、語及び語幹は明らかにデータベースに記憶されず、これらはこれらにアクセスするために使用されるキー順序を基にして構成されることになる。

【0097】日本語の場合、前記ツリーデータ構造には2つのタイプの命令が存在する。第1の命令は、語および句の発音と一致するかなの順序により構成されている語彙モジュールに蓄積された語及び句の読み方を作成する。各読み方に対応する各読み方と結合する見出し語を創作する第2の命令のリストがある。各読み方は直接先行するキー入力順序に結合された読み方の1つを変更する第1の命令により創作される。同様に、各見出し語は、第2の命令との結合を有する第1の命令により変更される読み方と結合された見出し語の1つを変更する第2の命令により創作される。

【0098】それぞれの第1の命令は既知のキーを参照するので、特別のかなが付随するような情報はキーと関連づけられたかなの組に論理的な索引として蓄積される。単一のキー540の典型的ダイアグラムが図12に示されている。具体例におけるキーの内部の論理的表現は物理的な配置を反映する必要はない。例えば541は日本語の語彙モジュールにおいて“「2か」”のキーに結合されたキーの好ましい論理的表記である。図12中の542に示されている追加された表記(き、く、け、こ)もまたキーに関連づけられているものである。また、これらの文字は図12の543に、日本語の辞書で使用する頻度が減少する(く、き、こ、け)の順番により適切に索引付けされる。図13に示されている表はキー上に表示された優先順位を表しており、日本語を入力確定に用いられる。拡張することにより、図13は日本語へのキープレスで曖昧でない状態で使用するためのキー索引についての論理記号索引に関する好ましい表である。

【0099】図13は図1に記載された好ましい実施形態に対応する表を示し、これは分離した区別されたキー68を有し、1つ前の文字に濁点か半濁点を入力するときに用いられる。区別されたキー68を押すことは、前のかに濁点か半濁点のどちらかを入力することは関して曖昧である。他の実施の形態においては、区別されたキー68は曖昧ではなく濁点を入力するには区別されたキー68を1回押し半濁点の時には連続して2回押す。また、他の方法では、(濁点がある文字やない文字を含めて)すべてのかなは決まった1つのキーに割り当てられており、文字表においての同じ行で示される。このことはデータベースを使用するシステムが区別されたキー68の使用を任意であるようにセットすることを認める。かかるシステムにおいては、区別されたキー68の使用を要求する選択がされていない状態では、区別されたキー68が1回(濁点)または2回(半濁点)押されるまでは、区別されたかなの追加を特定する命令は行われない。

【0100】ワードオブジェクト語彙モジュール1010の典型的な図が図20に示されている。データツリー構造は対応するキー入力順序を元にした語彙モジュールに変換候補を配置するのに使用される。図20に示すように、語彙モジュールツリーの各ノード、N001、N002、N011は典型的なキー入力順序を表している。ツリーに表示されているノードは、P001、P002、P011のパスでつながれている。曖昧でないシステムの実施の形態では、11の曖昧なデータキーが存在するので、語彙モジュールツリーにある各親ノードはさらに下の11の子ノードにつながれている。

【0101】パスでつながれているノードは有効なキー入力順序を示し、一方、ノードからのパスがないものは、無効のキー入力順序を示す。即ち、記憶された語の

いずれとも一致しないものなどである。注意すべき点は、無効なキー入力順序があった場合には、実施形態のシステムは入力されたキー順序の数字による表示の説明を作成し表示することができることである。文字の変換挿入点に、数字による表示の説明を(選択キーを押さずに)表示することは、入力キーに該当する単語が語彙モジュールに含まれていなかったことを示す。

【0102】語彙モジュールツリーは、受信したキー入力順序によってジグザグに変化する。例えば、ルートノード1011で第2のデータキーが押されると、第1のキーと結合したデータをルートノード1011の内部から取り出しそして評価することとなる。そうすると、パスP002を通してノードN002に移る。その後、再度、前記第2のデータキーを押すと第2のキーに結合されたデータがノードN002から取り出されそして評価されることになり、その後、ノードN102へのパスP102に移る。さらに、以下に詳細に示すように、各ノードはキー入力順序に対応する多数のオブジェクトと関連づけられる。キーが入力が受信されそして対応するノードが処理されると、キー入力順序に対応するオブジェクトに関するオブジェクトリストが生成される。各語彙モジュールのオブジェクトリストは、曖昧でないシステムの主ルーチンにより、選択リストを生成するために使用される。

【0103】図14は、各ノードに関連づけられたデータ構造400のブロック図である。前記データ構造は、語彙モジュールツリーでの各ノードとさらに下にある子ノードとのリンク情報を持つ。また、前記データ構造は、ノードのよって表される、典型的なキー入力順序に関連づけられたオブジェクトを識別するための情報(命令)も備えている。

【0104】ノードデータ構造400における最初のフィールドは、親ノードとつながっている子ノードの数と同一性を示す有効なキービットフィールド402であり、11の有効なキーの親ノードは、ノードで表された特定のキー入力順序に関連づけられたオブジェクトを識別(構成体)するための情報(命令)に結合されている。本発明の実施の形態においては、11のデータキーがあるので、多くとも11の子ノードはいずれかの親ノードとつながることができ、それ故に11の有効なキー情報は子ノードの有無を示すために有効なキー情報フィールドに提供される。各有効キー情報は、語彙モジュールにおけるそれぞれの子ノードデータ構造へのポインターを有するポインターフィールド404a、404b、・・・404nに関連づけられている。

【0105】子ノードは、子ノードに関連づけられたキー入力が親ノードに関連づけられたキー入力順序を有効に継続するものであるかどうかを、ただ表示するだけであるので、ポインターフィールドの数は各ノードに対し変化する。例えば、フィールド402の有効キー情報

10

20

30

40

50

は、可能な11個のキー入力の内6個のみが有効な子ノードに導くことを示す。何故ならば、6個の有効パスだけが存在するので、6個のポインターフィールドが親ノードに関するデータ構造に含まれる。フィールド402の有効キー情報は、ノードデータ構造内に含まれているポインターフィールドの認証を確認するのに使用される。もし、キー入力がある有効な子ノードへ導かなかった場合、関連づけられたポインターフィールドは、語彙モジュールを記憶するために必要とされるメモリ空間の量を保存するためのノードデータ構造から省かれる。

【0106】各ノードは、そのノードにより表現されるキー入力順序に対応する多数のオブジェクトに関連づけられる。各オブジェクトは、ノードデータ構造内に含まれる有効キービットフィールド402内のビットパターンにより示されるような、特定の有効キーに付属する図14のパケット408中の406内の命令により記述される。

【0107】各パケット406での各命令は、各ノードにより表現されたキー入力順序に相当するオブジェクトの1つを記述する。オブジェクトの記載は2つのオブジェクトリストの継続を要求する。図21では、語彙モジュールツリーの親と子から曖昧でないソフトウェア処理により動的に創作された代表的なオブジェクトリストを記載する。オブジェクトリスト430は、2回のキー入力力で表現されるノードに関連づけられたオブジェクト1-N₁を含むオブジェクトリストである。オブジェクトリスト440は、3回のキー入力力で表現されるノードに関連づけられたオブジェクト1-N₂を含むオブジェクトリストである。各オブジェクトリストは、各ノードに関連づけられたすべてのオブジェクトリストを有している。オブジェクトリスト430は、図1のキーボードよりキー入力順序「2か」、「2か」を表現する親ノードと関連付けられている。オブジェクトリスト440は、キー入力順序「2か」、「2か」、「2か」を表現する子ノードと関連付けられている。各ノードに関連づけられた実際のオブジェクト数を考慮してオブジェクトリストのサイズが変化することが認識できるであろう。

【0108】子ノードに関連づけられた各読み方のオブジェクトは、親ノードにより構成されたオブジェクトにかな文字を追加するという第1の命令を用いて構成されている。図14の命令406のパケット内の各命令は、それ故に、子ノードオブジェクトを構成するために使用されるオブジェクトを親ノードオブジェクトリストから認識する図15に示されるフィールド556のOBJECT-LIST-INDEXを有している。例えば、図21を参照すると、古いオブジェクトリスト430内の第3のオブジェクト「きか」は、新規なオブジェクトリスト440の2番目のオブジェクト「きかく」を形成するために使用される。前のオブジェクト識別フィールド556のOBJECT-LIST-INDEXはそれ故に新規なオブジェクトを形成する

ために使用される古いオブジェクトを識別するために古いオブジェクトリストへの登録へのリンクを提供する。

【0109】インストラクション558は、新規なオブジェクトを作るために認識されたオブジェクトに加える符号を示すため、フィールド555のLOGICAL-SYMBOL-INDEXを含む。LOGICAL-SYMBOL-INDEXフィールドは、それ故に、新規なオブジェクトを形成するために付加されるノードのキー順序内の最後のキーからの文字を明示する。文字は図13に示されたような表を用いて明示される。例えば、図21に示すように、新規なオブジェクトリスト440内の第1のオブジェクト「かかく」は、古いオブジェクトリスト430の第4のオブジェクト「かか」を使用し、「く」を明示する追加のキー入力を付け加えることによって形成される。図13の論理記号索引表において、「く」は「2か」のキーの2番目の論理文字であり、それ故に、オブジェクト「かかく」生成した命令の論理的記号索引表フィールドは、表の第2の文字を示すために2に設定される。この方法によるオブジェクトの符号化は、各語彙モジュールに必要な記憶スペースを大幅に削減するため、各ノードと既知のキーの文字の集合に結合された既知のキー順序を使用する。

【0110】語彙符号化技術はまた検索せずに語彙モジュールの登録にアクセスすることを認める。各新たな有効なキーが入力されると、このシステムは古いオブジェクトから新規なオブジェクトを形成するため現ノードにおけるキーに関連づけられた命令を実行し、さらに最適な子ノードのための単一のポインターが追従する。また、語彙モジュール内に全てのオブジェクトを記憶しなければならないよりは、新規なオブジェクトは古い解釈に追加するLOGICAL-SYMBOL-INDEXフィールドを使って規定される。このように、モジュールにある複数のオブジェクトにより共有される語幹は一度だけ記憶され、それからすべてのオブジェクトを創作するために使用される。開示された記憶方式は、子ノードのオブジェクトリストを形成のための語彙モジュールにおける親ノードからのオブジェクトリストを維持することを要求している。

【0111】図13に示すような論理記号索引表への登録は単一の記号を必要せず、1つの登録に任意の順序が使用することができる。例えば、語「きかなかった」を形成するために、古いオブジェクトリストから、かな順序「なかった」を第3のオブジェクト「きか」に追加する。このような方法では、入力されたキー入力順序の長さは、関連づけられたオブジェクトの長さとは直接一致する必要はない。符号索引表への登録として記憶された一連の文字は、任意のキー順序により特定されるべき語彙オブジェクトを認めることになろう。すなわち、語彙モジュールの中の随意的位置に蓄積されることになろう。

【0112】オブジェクト・タイプ・フィールドは構成

されたオブジェクトに関する追加情報を指定するためにまた図15の各命令558内に含まれる。オブジェクト・タイプ・フィールドは、生成されたオブジェクトが語、語幹、またはその他のオブジェクトかどうか特定する符号を含む。それ故に、オブジェクト・タイプ・フィールドは、与えられた語彙モジュール内で組合わされるオブジェクトの異なる型を認める。さらに、オブジェクト・タイプ・フィールドは、語の発音 (speech) の部分の情報または多数の変形と活用語尾を形成するのに必要な情報を保持している。小型化された曖昧でないキーボードシステムは、発音情報の部分を保持する語彙モジュールを用いることで、曖昧でない処理を改善するためにシンタックス解析を実行する付加情報を使用することができる。また、オブジェクト・タイプ・フィールドは、圧縮された形式におけるテキストの伝達のための固有のコードを保持することができる。前記固有のコードは、入力されたキー入力順序または関連づけられた曖昧でない文字を送信する代わりに遠方の端末装置に送信されるであろう。

【0113】望ましい語彙モジュールツリーデータ構造のキーの特徴の1つは、各ノードに対応したオブジェクトがそれらの使用頻度に基づいてノードデータストラクチャー400に記憶されることである。すなわち、命令406のパケット内の最初の命令によって構成されたオブジェクトは、第3の命令 (存在するならば) よりも高い使用頻度を有する406内の第2の命令 (存在するならば) により構成されるオブジェクトよりもより高い使用頻度を有する。この方法においては、オブジェクトはオブジェクトリストに自動的に配置され、使用頻度の減少に従って記録される。この記載の目的において、単語オブジェクトの使用頻度は使用する代表的な集積の中の所定の単語を用いる可能性に関係し、これはその集積内で各語が現われる時間数に比例する。語幹オブジェクトの場合においては、使用頻度は同じ語幹を持つ全ての単語の使用頻度の合計で決定される。

【0114】各ノードにおける使用頻度または他の分類情報を記憶することは、システムが使用されているときに、各オブジェクトの分類を決定しそして分類する必要性を回避する。このことは、蓄積されオブジェクトは多数のより長い語に共有する語幹を含むことができるので、語プロジェクト用語表において重要な関連がある。それらの語幹の関連する分類の決定は、子ノード全体のツリーと各語幹の蓄積情報の詳しい考察を動的に必要とし、これはポータブルコンピュータにとって大きな処理負担を追加する。前もってこの情報を決定しそしてそれを語彙データに格納することは、処理負担を減少させる。更に、使用頻度またはノードによって作成されたオブジェクトの分類が、それらを創作する命令406の命令により暗黙に表現され、この情報のための追加の記憶スペースは必要とされない。

【0115】好ましくはオブジェクトがそれらの使用頻度に従って順番にノードデータ構造400の中に蓄積されると同時に、使用フィールドの頻度が各命令に関係付けられることが認識されるであろう。使用フィールドの頻度は、関連するオブジェクトの使用頻度に対応する代表的な数を含むであろう。異なるオブジェクト間の使用頻度は各オブジェクトの使用フィールドの頻度を比較することにより決定されるであろう。各オブジェクトパケットを有する使用頻度フィールドを関連づけた、後の構成を使用する利点は、使用フィールドの頻度が曖昧でないシステムにより変更しうることである。例えば、このシステムは、一般的なテキスト入力を行っている間に、ユーザーが使用する語彙モジュールの中にある所定のオブジェクトの頻度を反映して、使用フィールドの頻度を変更することができる。

【0116】別の優位性としては、上述した自動活用 (auto-inflection) の実施形態のように、別のツリーからのそのオブジェクトが動的に一緒に格納できることである。

【0117】図22は、特別な語彙モジュール内の対応するオブジェクトを確認のため受信されたキー入力を分析するためのサブルーチン600のフローチャートである。サブルーチン600は、特別なキー入力順序のためのオブジェクトリストを構成する。フローチャート中のブロック602は新規なオブジェクトリストをクリアする。ブロック604は、そのルートノード1011で、図20に示すツリー1010の処理を開始する。ブロック606は最初のキー入力を受けることになる。ブロック608~612は全ての利用可能なキー入力を処理するためにループを形成する。

【0118】図27に示すように、ブロック608はサブルーチン620を呼び出す。決定ブロック610はあらゆるキー入力が処理されたかどうかを決定する。もし、いずれかのキー入力が処理されないままの場合は、ブロック614は完結したオブジェクトリストに戻る。もしメインルーチンが、それぞれが最後ではない1より多いキーを有し、最後ではない全てのキーが前の実行と同一であり、新しいキー入力順序で繰り返しサブルーチン600を呼び出した場合には、ブロック602および604の初期化はサブルーチン620が最も新しいキー入力のみを処理するため直接呼び出された場合には迂回することが適切であろう。

【0119】図27はサブルーチン600からサブルーチン620を呼び出すためのフローチャートである。上で説明したように、曖昧でないシステムは、新たなオブジェクトを構成するために、以前のオブジェクトリストのコピーを用いて処理を開始する。ブロック626ではそれ故に、それが新規なオブジェクトリストを組み立てるために使われるように、前のノードからのオブジェクトのリストが記憶される。

【0120】図7で示されたメインルーチンでは、キー入力はブロック150のシステムによって検出される。有効なパスがキー入力に対応する子ノードにあるときは、新たなキー入力の受信の結果により、辞書モジュールツリー内で下方への移動が生ずる。図27のブロック621において、命令406および404aのようなポインターフィールドから成る該当するパケット408が入力されたキー入力に対応して存在するかどうかを決定するため、該当するキービットフィールドの試験が行われる。もし、キー入力と一致する有効なパケットがないときは、ブロック624において、前のオブジェクトリストは、選択リストを生成するために、メインルーチンに返される。

【0121】受信されたキー入力は辞書モジュールの中のいかなるオブジェクトとも一致しない無効のキー入力順序の一部であるので、キー入力は無視され、そして現在のオブジェクトリストは、辞書モジュールからのオブジェクトリストであるとして、再びメインルーチンに戻される。従って、ブロック622及び624を含むサブルーチン620のブランチは、いずれの無効のキー入力順序をも無視し、曖昧でないシステムにより作られた選択リストに含ませるため親ノードにおいて生成されたオブジェクトリストに戻る。

【0122】決定ブロック622において、受信したキー入力に一致する該当するパケットが存在するならば、サブルーチンは前のオブジェクトリストに新規なオブジェクトリストがコピーされるブロック626に進む。ブロック628は与えられたキーに関連する最初の該当命令を取り出す。最初の命令が新規なオブジェクトリスト内で最初のアイテムを生成するように、ブロック630は繰り返しNEW-INDEXを1に初期化する。サブルーチンはそこで、該当する命令と関連するオブジェクトリストを組立てるため、ブロック632から642までを含むループに入る。ブロック632において、OBJECT-LIST-INDEX フィールド556はテストされ、そして対応するオブジェクトが前のオブジェクトリストからロードされる。ブロック634において、論理記号索引 (LOGICAL-SYMBOL-INDEX) フィールド555は調査され、そして適切な記号 (図13の550のような論理記号索引テーブルを経て受け取られたキーストロークと関連している) が識別されたオブジェクトの最後に付加される。

【0123】もし与えられたキー551および論理記号索引552における記号テーブル550への入力が文字順序を保持するならば、1より多い文字がブロック634において識別されたオブジェクトに付加されることが認識されるであろう。ブロック636において、関連づけられたオブジェクト及び記号は新たなオブジェクトリストの新規なオブジェクトとして蓄積される。ブロック638において、サブルーチンが所定のノードにおいて所定のキーと関連する最後の有効な命令を処理したかど

うかを決定するために試験が行われる。最後の有効な命令が処理されなかったならば、ブロック640では、次の有効な命令が取り出される。ブロック642では、NEW-INDEX が加算される。

【0124】決定ブロック638のテストがオブジェクトの全てがノードのために構成されたことを示すならば、サブルーチンはブロック644に進み、そして子ノードに関連するポインターに従う。ブロック646で新規なオブジェクトリストは選択リストを生成するためにメインルーチンに返される。各ノードと関連するオブジェクトリストを生成するための図22におけるサブルーチン600がユーザーから受け取られる各キース入力のために形成されるということが認識できるであろう。各キー入力は単に語彙モジュールツリーの中のサブルーチンをレベル1つだけ進めることになるので、語彙モジュールの“検索”はユーザーが新しいキースト入力順序を入力することで行われない。検索は各キース入力のために行われないので、語彙モジュールは最小の処理負荷を有する各ノードに関連するオブジェクトのリストを戻す。

【0125】従来の一般的な技術の内の1つで明らかにように、この技術で知られている他のフォーマットのデータベースは、ここで指定されたデータベースの増大または置き換えを行なうことができる。

【0126】V. 見出し語の読み方の変換をサポートするためのデータベース構造見出し語の読み方からの変換をサポートするために、各最初の命令 (この命令は、読み方のための平仮名の文を拡張する) は、ゼロまたはそれ以上の第2の命令のリストに関連づけられ、第1の命令によって拡張された読み方に関連づけられた見出し語を拡張する。第1と第2の命令の両方は、メモリ内の同じ物理的なツリーに格納される必要がない。命令の2つのセット (sets) は、第2の命令ツリーのノードと一致し、第2の命令セットの順序に一致における、最初の命令ツリーノードの最初の命令の順序 (sequence) まで、個別のツリーに蓄積することができる。同様に、第1の、そして、第2の命令が、メモリの同じ物理的ツリーに蓄積されるとき、それらは、各、最初の命令が、即座に一連の (ゼロ以上) の関連する第2の命令に従っているように、結合することができる。代りに、ノードにおける最初のすべての命令は、第1の命令と同じ順序における、接触しているブロックの第2の順序に従っている次の (contiguous) ブロックにおける順序、(ゼロ以上) の関連する第2の命令を含む各ブロックに蓄積することができる。

【0127】発明の更にもう一つの観点において、データベース圧縮における、更に大きい効率性は、特定の読み方 (reading) に関連づけられている、各漢字文字 (character) のみあらゆる特別な関連する読みのための、データベース構造に1度だけ蓄積することによって、達成され

る。一般的に、データベースは、同じ読み方の同じ漢字のいくつかの異なる例、例えば、動物（どうぶつ）と植物（しょくぶつ）においての、漢字、物（ぶつと読む）である。漢字文字(character)の様々な変わった(unusual)組み合わせの入力を容易にするために、データベースに明白に含まれない変わった名前、または、語(word)のように、例えば、各漢字は、所望の関連する読みでのみの入力で、特定されることができべきである。このように、望ましい実施の形態において、すぐツリー構造のルートからスタートする、与えられた漢字の各読みは、データベースに含まれ、第2の命令に付随して、関連した漢字のためのコードの仕様を含む。この命令は、同じ第2の命令リストにおける別の命令によって、同じ読み、そして、同じ特定(specified)として、関連づけられた、他のテキストオブジェクトに関連する、特別な漢字の存在の相対度数決定されたオーダにおける第2の命令リストにおいて、蓄積される。全てのそのようなケースにおいて、データベースツリーのルートで、読みのスタート(start)を関連づけることを構築した、ノード順序の後、一致する第2の命令は、漢字を無効のテキストオブジェクトに付加する。このように、漢字を無効のテキストオブジェクトに付加する全ての第2の命令には、特徴があるフォーマット(インデックスが前の見出し語のインデックスを特定するのに必要とされない)を有し、漢字文字コードを明確にするために用いられるビット数は、データベースにおける、表された有効な漢字の大多数の中で、十分に達成したり、識別したりすることができる。

【0128】相応じて、他よりも、ワード、または、フレーズの最初のキャラクタとして、データベースのどこかのワード、または、フレーズにおいて、漢字文字が表れたとき、関連する第2の命令は、第1のオブジェクトにおける一致する読み方のための第2のオブジェクトリストに、即座に先行しているノードの第1のオブジェクトリストにおいて、以前に加えられた現存する見出し語テキストオブジェクトに漢字を加える。このように、この場合においては、命令フォーマットは、第2の見出し語テキストオブジェクトのインデックスが特定されるフィールドを含まなければならない。同様に、最後のノードにおいて、漢字が追加されることができ、存在している見出し語が、「保存され」なくてははいけな

とき、または、かな（第1の命令において、明確にされた）が追加されなければならないときは、第2のオブジェクトインデックスフィールドは、命令に含まれていなければならない。このように、これらの2つのタイプの第2の命令のフォーマットであり、1つは、第2のオブジェクトインデックスではなく、他方は、そのようなインデックスであり、データベースのサイズを最小限にするために、区別(distinguished)であるべきである。

【0129】後のケースにおいて、すなわち、命令における第2のオブジェクトインデックスがnon-zeroであり、そして、漢字が見出し語に付加されるとき、付加された漢字に関連づけられた読みは、システムによって決定される。なぜなら、システムは、音節がキャラクタからの第1のオブジェクトリストにおいて、関連する読みに付加され、命令でのインデックスにより特定された見出し語テキストオブジェクトに、最後に追加されるトラックを保持するからである。上記で説明したように、各漢字の各読みは、ツリーのルートでスタートするデータベースに蓄積され、漢字のためのコードが関連づけられた命令において、十分に特定される。このように、「漢字が、以前に存在している見出し語に付加されている」という命令において、その漢字は、十分に特定された漢字、すなわち、ツリー構造ルートから、開始する決定された読み(reading)の第2のオブジェクトリストに表される漢字の位置に対応するインデックスを記憶させることによって特定される。この読みは、一般的に要求されることよりも、はるかに少数で特定される、データベースにおいて、漢字の存在の大多数を可能にし、蓄積されたインデックスは、データベースが作られた時に決定されることができ、かなり限られた値の範囲内において、消去されるであろう。

【0130】例えば、データベースにおいて、約89,000個のワードが記憶されており、漢字の変換の約89パーセントは、この方法で記憶される。

【0131】1つの実施の形態として、図16に示すように、4つのタイプの第2の命令が存在し、明確な4つの値の1つに相当する2ビットフォーマットフィールドセットを各々開始する。

【0132】例えば：

【表2】

00	保持指令5600：現在のノードによって生成されたリストの中の第2-オブジェクト-リスト-インデックス (SECONDARY-OBJECT-INDEX) によって特定された以前のリストから見出し語を保持する。(一次指令によって付加された仮名は、この見出し語に与えられる次の漢字の要素を特定するために用いられる読み方の一部となる。)
01	仮名指令5601：一次指令によって付加された仮名 (文章の仮名として) を、第2-オブジェクト-リスト-インデックス (SECONDARY-OBJECT-INDEX) によって特定された以前の見出し語に加え、この変更した語を現在のリストに加える。
10	非直接漢字 (INDIRECT KANJI) 指令5602：一次指令によって付加された仮名は、第2-オブジェクト-リスト-インデックス (SECONDARY-OBJECT-INDEX) によって特定された以前の見出し語に付加されるべき漢字の読み方を完成させる： 分割N-ビットフィールドの漢字-インデックス-フロム-ルート (KANJI-INDEX-FROM-ROOT) は、付加される漢字が、ルートノードから出発する完全な読み方を追跡することで、到達した現在ノード中の完全に特定された見出し語として発見された場合、そのリストインデックスを含む。
11	直接漢字 (DIRECT KANJI) 指令5603：続くN'-ビットフィールド漢字コード (KANJI-CODE) は漢字を (見出し語の第1の要素として) エンコーディングすることを完全に特定し、漢字に関連づけられた読み方は、ルートノードの仮名から関連する一次指令によって付加された仮名までの仮名を含む、読み方に付加されるすべての仮名を含む。

【0133】したがって、図16に示すように、フォーマットフィールド5610が00, 01, 10にセットされた指令5600~5602は、すべてが、直前のノードによって生成されたリストの中の既存の見出し語インデックスを特定するフィールド5611のセカンダリーオブジェクト-リスト-インデックス (SECONDARY-OBJECT-LIST-INDEX) を含む。「保持 (PRESERVE)」指令5600において、インデックスを付けられた見出し語はリスト中に維持され、それが次のノードの1若しくは2以上の指令によって修正されることができるようになっている。「仮名」指令5601では、インデックスを付けられた見出し語は、関連付けられた最初 (一次) の指令によって特定された仮名を付加することによって修正される。「間接的漢字」指令5602では、付加される漢字を呼び出すための仮名は、指令5602が関連づけ

られる最初 (一次) の指令と共に、先行する指令の順序から決定される。付加される漢字は「漢字-インデックス-フロム-ルート (根幹からの漢字インデックス)」 (KANJI-INDEX-FROM-ROOT) フィールド5613によって指定され、そしてそれは二次指令リストの中の指令のインデックスであり、一次指令に関連付けられており、ルートノード1011から開始される順序の中から決定される最後の仮名に付加される。「直接 (DIRECT) 漢字」指令5603には、所定の漢字のフルエンコーディングを再生するために十分な情報を含むより大きい「漢字コード」 (KANJI-CODE) 「ビットフィールド5614が含まれる。全ての指令フォーマット5600~5604は、また、所定の一次指令に関連付けられたグループの最後の指令を区別するのに寄与する「ストップフラグ」フィールド5612を含む。

【0134】これらの種々のケースの例が図32、27に示されており、データベースツリー構造の3つのノード中のある可能性のある指令リストの代表を示している。なお、この説明の便宜上、図32、27は一次指令をデコードされた形で示し、さらに、現実にはデータベース指令に格納される論理シンボルインデックス(図13に示す)ではなく、付加される実際の仮名を示している。また、同じく説明の便宜上、それぞれの命令によって作成されたリストオブジェクトは指令の右側の弓かっこ内に示されている。図32、27は、「2か」キー(例えば図3のキー122)の3回の連続した操作に対応するノードの、一次指令及び二次指令のリストを部分的に示したものである。左手カラムは、第1レベルノード(すなわち、ツリー構造のルートからスタートすること)のための指令を示し、順序の中の最初のキー操作の結果として生成されたテキストオブジェクトを示す。続く2つのカラムは、2番目と3番目のキー操作による指令を示す。リスト中の各一次指令は、Y1、・・・、Yn(「Y」は「読み方YOMIKATA」の頭文字)と表示されており、各一次指令は関連する二次指令のリストと共に示されており、各二次指令はそれぞれM1、・・・、Mn(Mは「見出し語MIDASHIGO」の頭文字)で示す。図32、27において、Y0とM0は、「null(空白)」テキストオブジェクトを表す。したがって、期待されるように、ルートレベル(レベル1)において一次指令のすべてがY0を特定し、そして、すべての二次指令がM0を特定する。したがって、このレベルで代表される二次指令は、「直接漢字」指令(すなわち、第2-オブジェクト-リスト-インデックスフィールド(SECONDARY-OBJECT-LIST-INDEX)は、含まれない)であり、そして、図32、27に「M0+X」で示される各漢字キャラクター「X」のためのキャラクターコードは、指令の「漢字コード」フィールドで完全に特定される。「直接漢字(DIRECTKANJI)」指令は単に見出し語の最初の文字として現れる漢字を特定するために使用されるだけであるので、このような指令がツリーの第1レベルの下(前)に現われる場合でさえ、前の保持指令は必要とされない。したがって、「M0+P」の形の指令は、それらは推論されるから、レベル1には現れず、そして「null「空の」」テキストオブジェクトを維持する必要がある。例えば、真中のコラム(レベル2)で、一次指令Y1は、前のレベル(「か」)の読み方Y1に仮名「く」を付加し、これによって、第2のレベルに読み方オブジェクト「かく」を生成する。5番目の二次指令M5は、直接漢字指令であり、したがって、ルートノードから現在のレベルまでに付加された全ての仮名を含む完全な読み「かく」に対応する。その指令は、また、文字「画」の正しい出力コードを決定するのに十分な情報を含んでいる。この同じ漢字文字は、レベル3の一次指令Y2と関連付けられた二

次指令M1によって生成されている見出し語の一部である。前記一次指令Y2は仮名「く」を先行するレベル(すなわち、「きか」)の読み方Y11に付加し、したがって、第3レベルの読み方オブジェクト「きかく」を生成する。この一次指令Y2に関連付けられた二次指令M1は、「M5+[M5]」と表示されている。この指令は、「間接漢字」指令であり、したがって、(図32、27に「M5+」として示された)5の値を特定する第2-オブジェクト-リスト-インデックス(SECONDARY-OBJECT-LIST-INDEX)フィールドを含む。これは、前のレベル2においてY11(一次指令のオブジェクト-リスト-インデックス(OBJECT-LIST-INDEX)フィールドによって参照される)と関連付けられる5番目の見出し語オブジェクトに言及する。このレベル2の見出し語オブジェクトは、一次指令Y11すなわち「Y9+か」とそれに関連付けられた第2の指令M5すなわち「M7+P」によって生成され、前のレベル(レベル1)の読み方Y2の指令M7によって生成され完全に特定された、見出し語オブジェクト「企」が現在のレベル(レベル2)のリスト中の第5の見出し語オブジェクトとして複製(保持)されることを示す。この「保持された」指令は、プレースホルダーとして寄与し、そして、このレベルにおいて加えられた仮名「か」が後の指令によって付加されるための漢字の読み方の始めであることを示す。これは、インデックスを付けられた漢字[M5]に関連付けられた読みがレベル3(かく)で保持された「か」にY2によって加えられた「く」をプラスしたものにすることが知られている場合、レベル3の「間接的漢字」命令(Y2:M1:「M5+[M5]」)で起こる。インデックス[M5]が完全に特定された漢字「画」に導かれる場合、ルートから始まる読み「かく」を調べることによって、レベル2のY1に導かれる。レベル1からの保持された見出し語にこれを加えることにより、完全に特定された見出し語「企画」が生成される。

【0135】他の好ましい実施形態として、他の方法が、「保持」指令(前述の実施形態におけるフォーマットコード00)により供給された機能を実行するために使用される。この変形実施形態では、図17に示すように、フォーマットコード00を有する代替指令フォーマット5604は、指令フォーマット5602に類似し、それは、前に存在している見出し語のインデックスを特定する、漢字-INDEX-FROM-ルート(KANJI-INDEXFROM-ROOT)フィールド5613と第2の-オブジェクト-リスト-インデックス(SECONDARY-OBJECT-LIST-INDEX)フィールド5611の両方を特定する。しかしながら、この変形実施形態において、前記第2-オブジェクト-リスト-インデックス(SECONDARY-OBJECT-LIST-INDEX)は直前のノードを参照するのではなく、現在のノードよりも、「仮名の数(NUMBER-OF-KANA)」フィールド5615によって特定されたノードの数よりも1つ多い数だけ前のノ

ードを参照する。付加されるための漢字に対応している読み方は、参照されたノードから現在のノードで参照された関連づけられた一次指令までの一次指令のチェーン（連鎖）をたどって戻ることによって決定される。このアプローチの利点は、中間のノードにおいて、分割された「保持」指令が要求されないことであり、データベースをよりコンパクトにすることができることである。不利な点は、以前のノードからの複数のオブジェクトリストがキーストロークの順序を計算するために維持されなければならないことであり、以前のノードからのたった 1 つのリストでは足りないことである。同様に、仮名指令 5 6 0 5 の代替形式では、第 2 のオブジェクトリスト-インデックス (SECONDARY-OBJECT-LIST-INDEX) は、「仮名の数 (NUMBER-OF-KANA)」フィールド 5 6 1 5 によって特定された数だけ現在のノードよりも先行するノードを参照する。1 又は 2 以上の仮名が付加される順序は、参照されたノードから現在のノードで参照された関連づけられた一次指令までの一次指令のチェーン（連鎖）をたどって戻ることによって決定される。

【0136】上記と同じ利点と不利な点が代替「保持」指令に適用される。当業者に明かなように、「フォーマットコード」フィールド、「第 2 オブジェクトリスト-インデックス」フィールド、及び「仮名の数 (NUMBER-OF-KANA)」フィールドは、様々な方法で組み合わせ可能であり、例えば、ホフマン (Huffman) 符号化法を使うことで、ストアされた指令においてより大きな圧縮を成し遂げることができる。

【0137】この発明の他の好ましい実施形態において、データベースの更なる圧縮が、次に述べるメカニズムを通じて実質的に追加の処理負荷なく成し遂げられる。多くの場合、一つの単語において読み方が非初期位置に現れる場合、所定の漢字に関連する読み方が変更される。そのようなケースのかなり多くの場合において、この読み方の変更は前の漢字の発音の共通音声効果の結果であって、そして、最初の漢字の最初の子音を非発生から発生へと変更することを引き起こさせ、あるいは、摩擦音 (f, v 等の子音) を破裂音 (p, b, t, d, k, g 等の子音) へ変更することを引き起こす。これは非初期 (non-initial) の漢字の読み方の最初の子音に濁点若しくは半濁点を付加することに対応する。上述されたデータ記憶及び検索のスキームにおいて、追加の項目がデータベースに生成され、ツリー構造の根幹（ルート）から始まるノードの順序において格納された代替読み方（濁点若しくは半濁点が付加されたもの）に関連付けられた完全にエンコードされた漢字を格納する。

【0138】望ましい実施形態においては、漢字が非初期位置にそのような変わりの読み方と共に現れた場合、非直接漢字 (INDIRECT KANJI) 指令がその変わりの読み方を参照して望ましい漢字を特定するのに用いられる。データベースの大きさを減らすために、参照された漢字

は、代替読み方に関連してルート（根幹）からは格納されておらず、しかし、通常の読み方（濁点若しくは判断濁点がついたもの）には関連付けられている。意図する漢字 n を特定するために、間接的な漢字インストラクションの第 2 オブジェクトリスト-インデックスフィールドに格納された値は、次の方法で修正される： N を代替の読み方に関連づけられた根幹から格納された漢字の数に代表させ、そして、 X を、その漢字がルートから通常の読み方（濁点若しくは半濁点がないもの）に関連付けられて格納されている場合、意図する漢字の位置に対応する第 2 オブジェクトリスト-インデックス値に代表させる。そして、「間接的漢字」指令の第 2 オブジェクトリスト (LIST)-インデックスフィールドを値 ($N+X$) にセットする。「間接的漢字」指令を処理する場合、システムは、最初、格納された漢字を代替読み方に関連してサーチし、そして、 N （フィールド値 ($N+X$) よりも小さい）の漢字だけがそこに格納されていることを決定する。関連付けられた読み方が濁点若しくは半濁点を有する仮名から始まることを決定し、対応する通常の読み方が生成され、漢字の決定された数 N がフィールド値 ($N+X$) から差し引かれ、そして意図する漢字は決定されたインデックス位置 X における決定された通常の読み方のもとに発見される。

【0139】VI. 活用された語（自動活用）の動的生成をサポートするためのデータベース構造

有効なステムと接尾辞のリストから語を動的に生成することなく、ユーザーの入力したいすべての語を満足にカバーすることは困難である。この入力のために、本発明の別の実施の形態は、1 以上のベースツリーを、語のルートについての情報に基づいて提供し、また、1 以上の接尾辞ツリーを、語の接尾辞についての情報提供する。これらのツリー中での命令に関連しているのが接尾辞分類 (class) 番号であり、それは、言語に正当なものと、生成されたワードが互いに分類され、かつ全体的に指定された語から分類されることを可能にする頻度とに、ルートワードと接尾辞語との間の組み合わせを制限する。

【0140】自動活用データベース中の各ツリーの構造とは、接尾辞分類および頻度情報の追加、ならびに異なったツリーの位置および接尾辞分類とツリーとの間の関係を示す付加的なテーブルを除き、「曖昧なキーストロークの曖昧さの解消をサポートするためのデータベース構造」に記載されているものである。

【0141】接尾辞分類／頻度の説明

この実施の形態では、ルートツリーまたは接尾辞ツリーからの 1 次または 2 次命令のそれぞれが接尾辞分類番号と頻度のリストに関連づけられている。各リストの入力の構造は、図 18 に示されている。ベースツリーについて、接尾辞分類番号とは、この命令が完成させるルートワードに対して適用されることのできる接尾辞の分類の

インデックスである。接尾辞ツリーに関して、接尾辞分類は、対応した接尾辞が属する分類を生み出す。1以上の分類/頻度は所定の命令に適合することができる。分類/頻度エントリのリストの終りは、ストップビットによって示される。

【0142】ルートワードに対するゼロの接尾辞分類は、ルートワードが接尾辞のない形態で終了することを示す。接尾辞/頻度ペアのリストは、この分類（あるいは任意の他の分類）が常にリストの最後にくるようにして分けられ、この分類に対するエントリのためにストップビットを確保してもよい。

【0143】いくつかの命令は、ルートワードまたは接尾辞を完成させず、したがって接尾辞分類を全く有しない。本発明の好ましい実施の形態では、1ビットは、分類/頻度リストが後続するか否かを示す各1次または2次命令に関連している。別の実施の形態は、“接尾辞のない分類”という意味の接尾辞分類コードを与えることによって、これをエンコードする。前記頻度は、このコードのために省略することができる。

【0144】分類/頻度規制子のリストを有する代わりに、1次命令および2次命令の一方または両方に対するエントリが多くても1つ存在する可能性がある。そのようにして、ストップビットは使用が回避されるが、命令が1以上の分類以上に該当する場合にはば、それは複製される必要がある。明らかに、接尾辞分類番号および頻度の少なくとも一方にホフマンエンコードを実施し、必要とされるスペースを減少することができる。ただ一つの接尾辞分類、またはただ一つの頻度を有するツリー中では、前記フィールドが省略可能である。接尾辞ツリーに少数の接尾辞分類しか記憶されていない場合、そのツリー内の接尾辞番号の表示が短縮されてもよい。また、別の実施の形態は、可能な接尾辞を示すビットマップとして、命令が該当する接尾辞分類を示している。これらの分類と関連した頻度は、リストすることができ、または、1つの混合された頻度はこれが該当するすべての接尾辞分類に対して記憶されることが好ましい。

【0145】1次命令に対する分類または頻度規制子は、データベースから削除され、その代りにすべての関連づけられた頻度の合計と、関連づけられた2次命令のすべての分類の組み合わせから動的に計算可能なことが理解されるであろう。

【0146】接尾辞が付された読み方の生成活用のあるワードのリストは、以下（図23を参照）のように接尾辞ツリーとベースとから生成可能である。試みは、1つの文字からキー入力の数Nまでのルートワードのあらゆる可能性のある長さMに対してなされる。最初に、Nはキー入力の長さに設定され、Mは1（ブロック800）に設定される。Mの各値に対して、最初のMキー入力に一致するすべてのルートワードは、ベースツリー（ブロック802）から抽出される。同様に、残り

のキーのすべての可能な接尾辞の解釈は、同じアルゴリズム（ブロック804）で各接尾辞ツリーから抽出される。このとき、ルートワードと、分類が一致した各接尾辞との各組合せが生成され（ブロック806）、正当な完成されたワード（ブロック808）のリストに加えられる。ブロック810で、MがNと一致しないとき、Mは1つづつ増大され（ブロック812）、プロセスはブロック802へ戻り、それ自身反復する。ブロック810で、M=Nとして決定されたときは、図24及び図25に記述され、以下に詳しく説明されているように、プロセスは終了し、Nキー入力の完成したワードはソーティングの準備が整う（ブロック814）。

【0147】当業者によって容易に認識されることができるよう、このサーチ全体は、あらゆるキー入力に対して行われる必要がない。例えば、キー入力が増加されたときに、各ルートの長さに対して可能なルートワードのリストは変化せず、したがってシステムは、各ルートの長さに対して生成されたルートのリストを想起するだけでよく、追加されたキー入力を含む長いルートのリストをそれに加えるだけである。同様に、各接尾辞ツリー中の可能な接尾辞のリストは、以下にさらに詳細に記述されているように、スクラッチ（scratch）から文字が作られる代りに、図27で記述されているように、新しいキー入力に対して更新されることができる。

【0148】自動活用（Auto-Inflection）のための頻度生成

自動活用システムで生成されたオブジェクトの適正な相対的配列（order）を決定するために、頻度は、ベースおよび接尾辞ツリーに記憶された頻度値から作られた各オブジェクトに対して生成される。活用していない（接尾辞が付されていない）オブジェクトのベースツリーにおける頻度値は、オブジェクトが典型的な言語資料（corpus）中に認められた総頻度を反映している。活用したオブジェクトのルートのベースツリー中の頻度値は、任意の接尾辞を有する所定の接尾辞分類のメンバーとしてそのルートが使用された頻度をその分類から識別する。接尾辞ツリー中の頻度値は、この特定の接尾辞が所定の接尾辞分類中の全ての接尾辞と比較して使用される相対頻度を示す。

【0149】図24を参照すると、接尾辞を付けられた（活用する）オブジェクトに対する頻度を生成するために、システムはオブジェクトがベースツリーにおいて完全か否かを最初に決定する（ブロック836）。否の場合、システムは接尾辞ツリー中の接尾辞の頻度値によってベースツリー中のルートの頻度値を乗算する（ブロック840）。決定ブロック836で、ベースツリーにおいてオブジェクトが完全ならば、決定ブロック838において、オブジェクトが接尾辞が付されないで表すことができるルートワードであるか否かが決定される。イエスの場合、接尾辞なしで表すことができるルートワード

は、予め定められた値を掛けられたその頻度値を有し、接尾辞なしで表されるルートの相対頻度を生成する（ブロック 844）。他方、接尾辞を付さずに表することができない完全なワードは、それらの頻度が直接ベースツリーに記憶される（ブロック 842）。以下、図 25 を参照して、このようにして決定された種々の頻度のあいだの比較を説明する。

【0150】本発明に 1 つの実施の形態では、頻度または頻度値（使用された互換性のある）の対数が頻度の代りに記憶される。これは、頻度が重要な情報を失うことなくごく少数のビットで記憶されることを可能にし、2 つの頻度間の積がそれらの対数の和と置換されることを可能にする。さらに、無活用のルートに対する対数および接尾辞オブジェクトは、各々がそれを記憶するために割り当てられたビットで利用できるダイナミックレンジを正確に満たすように別々に定められている。例えば、ステム頻度が 32 ～ 1023 で変動する場合、これらの対数（ベース 2）は 5 からちょうど 10 未満まで変動する。4 つのビットに適応させるために、測定された対数が 0 ～ 15 から変動するように、対数は 5 を減算して、16 / (10 - 5) により乗算することによって再測定される。オフセットおよびスケールは記憶され、その対数が別の対数と比較される前にその対数を共通の範囲およびスケールに戻すために使用される。

【0151】自動活用選択リストのソーティング
図 7 のブロック 160 からのプロセスで説明したように、図 23 で述べられた自動活用のプロセスによって見いだされた結果は、一番使用頻度の高いワードを最初にし、相対的な使用頻度に従ってソートされなければならない。

【0152】自動活用選択リストは、図 25 に示された比較に従って、（従来の技術で知られたソートアルゴリズムを使って）ソートされる。ブロック 860 において、2 つのオブジェクトが異なった頻度（上述したように、対数または別のフォーマットで）を有していると決定された場合、それらの相対的な配列がそれらの頻度の基礎とされる（ブロック 862）。別な場合では、ルートワードが異なる長さである場合（ブロック 864）、最も長いルートを有する完全なワードが最初にくる（ブロック 866）。ルートワードが同じ長さである場合、ブロック 870 において 2 つのオブジェクトに対するルートワードが同じであるか否かが決定される。否ならば、ブロック 876 において、2 つの異なったルートワードのベースツリーにおける配列が順序（order）を決定する。ルートワードが同じならば、ブロック 872 において、同じルートを有する接尾辞の接尾辞ツリーにおける配列が順序（order）を決定する。

【0153】自動活用オブジェクトの漢字への変換
ユーザーが変換キーを押した後、ユーザが選んだ読み方と一致するいくつかのルート/接尾辞のペアが存在する

可能性がある。例えば、5-かな読み方は、同じかなに関して 2-かなルートおよび 3-かな接尾辞として解釈可能であると同時に 3-かなルートおよび 2-かな接尾辞として解釈できる可能性がある。ルートの同じ長さに対してさえ、同じ読み方は、多数の分類からのルートおよび接尾辞のペアに対応する可能性がある。変換リストは、各ルート/接尾辞ペアに一致するすべての見出し語を含むこととなる。

【0154】所定のルートの長さおよび所定の分類のルート/接尾辞ペアについて、ルートを完成させる 1 つの命令と、接尾辞を完成させる 1 つの命令とが存在する。接尾辞を完成する命令に明らかに関連した見出し語がない場合、読み方と一致する見出し語が読み方と同じ頻度で生成される。同様に、読み方に一致する見出し語は、ルートワードが別の見出し語を持っていない場合、そのルートワードを完成させる命令に対して生成される。変換リストは、接尾辞の見出し語とルートワードの見出し語のすべての組み合わせから構成されている。変換リスト中の最後の項目は、常に、最少頻度の接尾辞見出し語が後続するルートワードのカタカナ表示である。

【0155】自動活用変換リストのソーティング
変換リストの項目は、1 以上の命令から構成されているので、それらはまたソートされることが必要である。頻度は、図 24 に記載されたように計算される。このとき、1 次命令に関連した頻度を計算した場合とは異なり、2 次命令に関連した頻度が計算される。

【0156】自動活用変換リストは、図 26 に示されている比較に従って、（標準的な技術のソートアルゴリズムを使って）ソートされる。ブロック 880 では、2 以上の変換オブジェクト（見出し語）が、同じ頻度を持っているかどうか決定される。2 つのオブジェクトが異なる頻度（上述したような対数または他のフォーマット）を有している場合、それは、それらの相対的な配列（order）であると決定される。すなわち、頻度のより高い見出し語を最初に持ってくる（ブロック 882）。別の方法では、ブロック 884 において 2 つのオブジェクトに対して関連した 1 次命令が異なるかどうか決定される。そうある場合、これらは、それらの命令があたかも独立した読み方を与えるかのように、図 25 で与えられたアルゴリズムに基づいて検索される（ブロック 886）。そうでなければ、ブロック 890 において、2 つのルート見出し語が同じ 2 次命令からのものかどうかを決定する。もし、そうでなければ、ブロック 896 において、2 つの異なったベースの第 2 の命令のベースツリーにおける配列が順序（order）を決定する。イエスであれば、ブロック 892 において、接尾辞の 2 次命令の接尾辞ツリーにおける配列（order）が順序（order）を決定する。

【0157】VII. 言語モジュールツリーの圧縮
処理時間を増加することなく語彙モジュールに対して少

ない記憶容量を使用しながら、オブジェクトのより広い語彙を検索する新しいいくつかの手段は、図23、27に示されているように、オブジェクト識別ソフトウェアプロセスへのキーストロークの組合せに固有であり、それらの各ツリーデータ構造上で動作する。

【0158】入力語彙目録におけるそれらの使用頻度に従って所定の語彙モジュールの論理的記号インデックステーブル550 (図13)の各行に記号を配列することによって、ツリーデータ構造1010中のすべてのノード400の命令558 (図15)のほとんど大部分に、1と等しいそれらの論理的記号インデックスフィールド555を持たせることができる。同様に、文体 (language) におけるそれらの配列の使用を減少する際にオブジェクトリスト440 (図21)において、ステムおよびワードオブジェクトが生成されるように、すべてのノード400においてすべての命令パケット406の命令558を配列することによって、ツリー構造1010中のすべてのノード400のほとんど大部分の命令558に、1と等しいそれらのオブジェクトリストインデックスフィールド556を持たせることができる。このように、ツリー1010におけるデータの多くが重畳である。分類的 (Systematically) に、重畳を識別し、親ノードを子ノードにリンクするパスを再度方向付けて、これ以上参照することのない子ノードを削除することによってそれらを除去することにより、元のツリーよりはるかに少ないノード、はるかに少ない命令およびはるかに少ないリンクを含んでいる、さらに高度に畳まれ、あるいはラップされたデータ構造となるが、このデータ構造は依然として、元のツリーから検索可能なあらゆるオブジェクトを検索する。さらに、命令がオブジェクトリスト440における類似したオブジェクトを生成するオリジナルツリーを通るパスの別個の (distinct) インスタンスは、畳まれたツリー中の共通パスに合流させられ、それらは、以降汎用 (特殊と反対の) オブジェクト構成ルールとして機能し、それによって、減少した構造は、当初所定の語彙モジュールのツリー1010を規定するために使用された場合よりもはるかに多くのオブジェクトを生成することが可能となる。例えば、30000の英単語のリストから発生する開いた語彙ツリーは、実施形態の78000を越す命令を含むかもしれない。図22のフローチャートの好ましい方法に表現されているあいまいなキー入力シーケンスおよび検索方法の場合では、上記の折りたたみ方法の実施形態による折りたたみの後、変更された (modified) ツリーが含んでる命令は、構造が検索することのできるワードオブジェクトの数よりも少ない29000個よりも少ない可能性がある。各命令は、キー押圧に回答して単一の記号を追加することによって、図21のオブジェクトリスト430中のオブジェクトを1つだけ変えるので、このことは注目

に値する新しい結果である。これは、一般的なオブジェクト構成ルールとして命令の共有シーケンスを再使用するソフトウェア検索プロセスと畳まれたツリーの結果である。変更されたツリー構造のさらに別の特徴は、キーストロークにオブジェクトを関連付けるために生成されたルールの自動同定 (automatic identification) である。このようなルールを用いると、語彙モジュールは、高い成功率で、最初にそれを生成する時に使用されなかったワードおよびワードステムオブジェクトとキーストロークを関連づけることができる。例えば、語彙モジュールを作るための入力ワードリストは、ワード "sun", "run", および "running" を含む可能性があるが、ワード "sunning" は含まない。しかし、アルゴリズムによって生成された畳まれたツリー構造は、依然として対応したキーシーケンスに対する選択リストにおけるオブジェクトの1つとしてワード "sunning" を生成するであろう。

【0159】図19に示されているノードが例である。ノード560は、その有効キーフィールド562 "01010000000" における "1" で示されている、2つの有効キーを持っている。1実施形態では、"1" の位置は、第2及び第4のキーが有効パスであり、命令パケット及びこれらの566及び568とのそれぞれに関連づけられた子ノードへのポインター (識別子) とを有することを示している。パケット566は、子ノードにノード560をリンクしているポインター "P" が後続する3つの命令、"(1, 1, 0)", "(1, 2, 0)" 及び "(2, 1, 1)" を持っている。図22のサブルーチン600が、ノード560に達するキー入力のリストを処理した場合、図27のサブルーチン620が呼出され、"2" キーを処理する。このキーは実施形態では "2か" キーであり、次のことが起こりうる。命令561は、インデックス1において新しいオブジェクトを構築するために、インデックス1における古いオブジェクトに、"2か" ("か") キーの第1の論理的な記号を追加する。命令561の第3のフィールドである、"0" は、ストップフラッグ557 (図15)の間違った値であり、これが現在のパケットの最後の命令ではないことを示すので、次の命令563が解釈される。命令563は、インデックス2において新しいオブジェクトを作るために、インデックス2における古いオブジェクトに、"2か" (図13での "か") キーの第1の論理的な記号を追加する。構築されている新しいオブジェクトのインデックスは、命令自身の配列 (order) に内在し、新しいオブジェクトのインデックスは2である。例えば、第2の命令は常に第2のオブジェクトを構築する。命令563の第3のフィールドである、"0" は、ストップフラッグ557の誤った値であり、次の命令567が解釈される。命令567は、インデックス3において新しいオブジェクトを構築するためにインデックス1における古いオブジェクトに (図13で

の”く”)キー”2か”の第2の論理的な記号を追加する。命令567の第3のフィールドである、”1”は、これが現在のパケットの最後の命令であることを示すSTOP-FLAG 557の真の値であるため、サブルーチン620(図13)の解釈はブロック638からブロック644に移行することとなる。

【0160】独立した多数のノードとして、同じ目的にかなうことができる単独なノードの中に、異なる命令パケット408を含んでいる、2以上のノードを結合することができ、語彙ツリー1010での一定の(certain)ノードが、新しい意味において(in a novel sense)重複であるということの意味する。本発明の目的のために、ワード”redundant”は、図28-31で示されている実施形態を参照して、以下に記載されるであろう、ソフトウェアプロセスの制御によって、1つのノードが省く(dispenes)ことができる、ある意味では、2つのノードとの関係を用いる。

【0161】例えば、図19では、ノード574をノード560に例える。ノード574のキー”4た”上の命令パケット566と571は、正確に一致し、一方、ノード560のキー”4た”上の命令570は、ノード574のキー”4た”上の命令572と相反するため、他の作業をすることができず、両方の作業をするであろう、1つに結合された2つのノードをすることができない。ノード560をノード576に例える。命令パケット566と577は、各ノード上で正確に一致したキー”2か”に関連づけられている。命令569と578は、それらのストップフラッグフィールド557のセッティングで異なっており、その違いはそれらに相反を生じる。図27でのサブルーチン620のオブジェクト検索プロセスの本質的な結果は、与えられたキーのためのノードについての命令セットを実行することによって作られた、新規のオブジェクトリストである。付加的な(Additional)オブジェクトは、そのノードの複数の子ノードのいくつかの正しい処理の決定なしに、オブジェクトリストの最後に追加することができる。このように、エラーは、ノード576の複数の子ノードの処理において、578の後の付加的な命令の実行に起因するであろう。プロセスの本質は、もし、間違っただけの場合、または、ほんの少しの命令が実行されてしまった場合だけに阻止することである。同様に、ノード576のキー”9ら”上の有効キーの存在は、ノード560での有効キー”9ら”の存在しないことと相反しない。ゆえに、ノード560と576とは重複であり、また、両方のネット効果を達成した、新たなノード582と合併することができ、両方の複数の子ノードの親ノードとして正しく機能する。ポインターは、また、確定した(defining)重複としての役割を果たすということが知られる。さらに長いワードからステムが形成されることを継続しないワードと関連した、ツリーに

おいてのシーケンスの最後のキー入力では、有効キーパケット408においてのポインターは、特別な値、”NULL”を持っており、実施形態において、さらに複数の子ノードが存在しないということを示す。そのようなノードは、”ターミナルノード”と呼ばれている。両ノードに共通している有効キー上に子ノードを有する2つのノードについて、端末ノードに達するまで、あるいは比較されているノードに共通の有効キーシーケンス上にノードの子孫が1つも存在しなくなるまで、各子ノードはそれらの親ノードに対して重複でなければならない、これら親ノードはそれら子ノードから派生するノードに対して重複でなければならない等、同様のことが続く。

【0162】図28-31は、1010と同様な畳まれた語彙モジュールツリーと圧縮のための、ソフトウェア処理の好ましい実施形態のフローチャートを示している。図28は、圧縮された語彙モジュールを構成するソフトウェア処理の好ましい実施形態のフローチャートである。ブロック652では、語彙目録は、日本語語彙モジュールのための図12のように、物理キー上の表示されているそれらよりも別の、いくつかの必要な付加的で曖昧な記号を特定するために調べられる。図13の例のように、ブロック654-656では、記号が語彙目録に入力されるとき、それらの使用頻度の減少順序において、それらの各キーに、それらの論理的な複数のインデックスが割り当てられる。頻度情報を持ったオブジェクトの語彙目録であるとする、同業者に明らかな従来の技術では、フォーム1010の語彙ツリーを構成する。ブロック660では、重複ノードは、識別され、データの重複を最小限にするために結合され、このことによって、一つのオブジェクトに関連する命令シーケンスを分離する順番(turn)は、多数のオブジェクトを検索するルールを一般化する。図29で、このプロセスを詳細に示す。ブロック662は、ターミナルノードからのすべての残っている複数のヌル(NULL)ポインターを識別し、親ノードの最大数を有するノードの方に向くようにそれらを変更するようにして、モジュールでのルールの数を増やす。別のルールが、子ノードをヌルポインターに割り当てて適用できること、また、処理中のキー入力に関係した要因に基づいた、オブジェクト検索のときに、そのようなルールは、動的に適用することができることは認識されるであろう。ブロック664では、それらがスペースを節約するために、より高い頻度命令およびアドレスに割り当てられているより短いビット・パターンを有するビットの固有パターンとしてコード化できるので、各々の固有の命令558(図15)およびポインター404a(図14)の残留している例はカウントされる。実施形態は、従来の知られた技術であるホフマンコードである。さらに、多数の親ノードの子ノードであるノードは、高速検索を容易にするために、また、それらを配置するための必要なビットの数

を最小限にするために、特別な配列 (ordering) で記憶することができる。オブジェクトを記載するようにという命令558を記憶するように選ぶ際に、ブロック658でツリーを生成すると共に、オブジェクトがワードまたはワードのステムであるときに、それらのキャラクターシーケンスが都合よくツリー1010のノードの重複を増やすために用いてもよい追加のデータを含むと認められる。例えば、日本語の仮名の全てのペアが等しく共通であるというわけではなくて、例えば、「ま」の後に普通は「す」が続く。仮名ペアの統計、またはビグラム

(bigrams) は、前の仮名からのオブジェクトの一番適当な次の仮名を予測するために使うことができる。この種の予測については、図13の論理記号インデックステーブル550のあいまいな記号の論理的配列は、更に第1の位置の使用を最適化するために動的に変更することができる。予測は、仮名三つ組みまで3文字の銘、通常、n-グラム (n-grams) まで広げることができる。図29は、語彙モジュールのツリー1010を畳んだソフトウェア処理の実施形態のフローチャートである。ブロック670は、図28のフローチャートのブロック660から始まる。ブロック672は、ルートノード1011の後、図20のうちの1010の第1のノードから始まるために処理を初期化する。ブロック674は、ノードの位置を決めるために、図30のフローチャートの実施形態において表されるサブルーチン690を呼び出し、いずれである場合でも、それは、現在のノードで最大限に重複している。オブジェクトノードが分かる場合、決定ブロック676が重複したノード (node(s)) が一緒に合併されるブロック678に処理を命令し、オブジェクトにキー入力シーケンスに関連づけすることに
30 対する一般的なルールである共有されたシーケンスに、個々の例の複数を包含しているツリーからの二重のデータを除去する。決定ブロック676が失敗する場合、決定ブロック680は処理がされるかどうかをテストする。より多くのノードを処理する場合、フローは他のノードを識別するためにブロック682へ進む。

【0163】図30は、ツリー1010において与えられたノードに関する重複の最も高い程度を有するノードであると認める、ソフトウェア処理の実施形態のフローチャートである。ブロック690は、図29のフローチャートのブロック674から始まっている。ブロック692は、測定された重複のためのプレースホルダー、マックスセービング (MAX-SAVINGS) を初期化する。ブロック694は、図20のツリー1010のルートノード1011から始まるために処理を初期化する。ブロック696は、現在のノードに関して与えられたノードの重複を計算するために、図31の実施形態のフローチャートにおいて表されるサブルーチン710を呼び出す。決定ブロック698は、マックスセービングより大きい重複の程度が報告されたかどうかをテストする。もしそうな

らば、ブロック700は、マックスセービングとして与えられたノードに関して、最も重複していることが仮に分かるノード、および重複の報告された計測の識別ベストノード (BEST-NODE) を記録する。決定ブロック702は、全てのノードが評価されたかどうかをテストする。そうでない場合には、フローは、ブロック704に続き、それは、現在のノードから次のノードまで進む。ブロック704から、フローはブロック696に戻る。決定ブロック702でのテストの結果が、最後のノードが評価されたということである場合、ともかく、図29のサブルーチン670のブロック674に対する与えられたノードに、ブロック706は最も重複したノードの識別の証明を返す。図31は、2つの指定されたノード間の数の重複を計算するソフトウェア処理の実施形態のフローチャートである。ブロック710は、図30のサブルーチン690のフローチャートのブロック696から始まる。ブロック712は、二重の命令のカウントを初期化する。ブロック714は、1までキーインデックス (キーインデックス) を初期化する。ブロック716は、サブルーチンにパラメータとして特定される2つのノード400の中で、第1のものから重要なキーインデックスと関連する命令パケット406を読み込んで、一時的なリストであるリスト-Aにそれらを記憶しておく。重要なキーインデックスが有効なキーでない場合、命令は読み込まれない。ブロック718は、サブルーチンにパラメータとして特定される2つのノード400の中で、第2のものから重要なキーインデックスと関連する命令パケット406を読み込んで、一時的なリストであるリスト-Bにそれらを記憶しておく。重要なキーインデックスが有効なキーでない場合、命令は読み込まれない。決定ブロック720は、どちらかのリスト-Aかリスト-Bが空かどうか決定する。そうでない場合には、ブロック722はリスト-Aおよびリスト-Bから1つの命令を呼び出し、そして、1までに各々の残っている命令の数を減少させる。決定ブロック724は、それらのLOGICAL-SYMBOL-INDEXとOBJECT-LIST-INDEXで、命令が同じであるかどうかのテストを行なう。そうでない場合には、重複のための失敗コードは、図30のサブルーチン690のブロック696に返されない。ブロック724の決定がイエスである場合、ブロック728は、カウントSAVED-INSTRUCTIONSを増加させる。ブロック720に対して進むために制御する。決定ブロック720がTRUEをテストする (イエスである) 場合、制御は決定ブロック730に進み、そして、それは2つのノードが全ての可能なキーに関して比較されたかどうかをテストする。そうでない場合には、ブロック732は、キーインデックスを増加させ、そして、制御はブロック716に進む。決定ブロック730がイエスである場合、1までキーインデックスをリセットするブロック734に進む。ブロック736は、2つのノードの重要なキーイ

10

20

30

40

50

ンデックスと関連するポインターを調べる。決定ブロック738は、どちらかのポインターが空(NULL)かどうかテストする。そして、それはターミナルのノードのポインターのための、または、有効でないいかなるキーのための事例(case)である。どちらのポインターも空でない場合、制御はブロック740に進む。そして、それは2つの空でないポインターによって示されている子ノードが重複しているかどうか、再帰的にテストするためにサブルーチン710を使用する。ブロック740での結果は、決定ブロック742でテストされる。2つの子ノードが重複していないとわかる場合、失敗コードが返される。一方、2つの子ノードが、ブロック746によって蓄積される特定の数のスコアについては重複していると認識される。決定ブロック748は、最後のキー(好ましい実施形態のキー9)と関連するポインターがテストされたかどうかテストする。そうでない場合には、ブロック752は、キーインデックスを増加させ、ブロック736に進む。ブロック748でのテストが、全てのポインターがテストされたと判断した場合、2つのノードの重複の蓄積された数の計測は、元々、サブルーチンがブロック710でいつ始まるかについての識別を返す。

【0164】重複数値の算出は、各々のノードに存在する分岐の数、および子としてノードを示している親ノードの数のような、付加的な要因のカウントに加算するためのウェイトを与えることができると認められる。特定のキーと関連する命令の配列のために、2つのノードが重複していない場合、語彙目録に入力された低頻度ワードと関連する命令の配列が、より高い頻度オブジェクトと関連する命令の優先権に影響を与えずに再配列することができると認められ、したがって、ツリーの重複を増加させる。

【0165】重複したノードを除去することによって、ツリーのサイズを減らすために上記に開示した方法は、1次命令から成るツリーに関して記載されていたことを明らかにした。1次命令は、2次命令に関連し、分類または頻度情報は、合併されるオブジェクトの複雑さを増すが、開示される手順(procedure)は、まだより複雑なツリーの重複を識別して、除去するために用いることができる。それらの1次命令に関しての重複しているノードは、合併することができ、関連する2次命令の少なくとも2つの別々の一組は合併することができる。そのとき、合併されたノードへジャンプする各々の親ノードでは、2次命令の代替の一組のうちどちらが実行されることになっているかを特定することは、少量の追加情報を加えるのにもつぱら必要である。また、もし、第2の命令が独立したツリーに記憶されているならば、第1の命令と第2の命令間の結合が全ての入力ワードのために維持される限り、2つのツリーはそれぞれに圧縮されることができる。

【0166】VIII. あいまいなキーストロークを使用しても作動するシステムの動作

図34は、図1に示されるシステムのオペレーションの4つの典型的な例を示すものである。これらの例は、システムの動作を説明し、そして、データキー21-30、セレクトキー60、コンバートキー62、区別的発音符キー68の使用を含む好ましい実施形態において提供されたフィードバックの性質を説明する。この例は、各キーストロークの結果としてテキスト領域66に現れるであろうテキストを示すものであり、そのテキストは挿入点88(点線下線若しくは実線下線)におけるオブジェクトの特殊なフォーマットを含む。図1に示すように、各キーストロークはキーの数ラベルによって特定される。

【0167】図34の例は、図1のシステムの好ましい1実施形態例において、フレーズ「お願いします」がどのように入力されるかを説明するものである。キー21、25、22に対する最初の3つのキーストロークのそれぞれの後で、最も頻繁に使用される読み方が表示され、それぞれ1、2、3の長さの仮名に対応する。区別的発音符(Diacritic)キー68に対する4番目のキーストロークはキー22に対する以前のキーストロークの仮名を濁点を有する仮名に特定し、そして、表示された読み方を最も共通する3つの仮名からなる読み方であってキー順序21、25、22に対応する読み方に変更し、三番目の位置に濁点を付す(うなぎ)。キー21に対する5番目のキーストロークの後では、最も共通する読み方は「おねがい」であり、これは好ましい入力フレーズの最初の語に対応する。

【0168】変換キー62に対する次のキーストロークは、表示されたテキストを最も共通する見出し語であって表示された読み方に対応する見出し語に変更し、それは、このケースではまた望ましい入力フレーズの最初の語に対応する。次のキーストロークはデータキー23に対するものであり、それはセレクトキー60或いはコンバートキー62に続く操作(このケースでは変換キー62に対する操作)であるため、それまでと区別されるべき新たなキーストロークの順序の入力を開始させるものである。キー23に対するこのキーストロークに引き続き、キー27に対する次の2つのキーストロークそしてキー23を再び操作することで、最も頻度の高い読み方が常に表示される。このケースでは、また、入力フレーズの望ましい語「します」に対応するそれらのキーストロークの後にその語が表示される。選択キー60に対する最後のキーストロークは、現在の語オブジェクトのキーストローク順序が終了したことを示し、それによって、キー21~30に対する次のキーストロークが新たな入力順序をスタートさせる。

【0169】図34における例2は、データベースに含まれているフレーズ「ありがとうございます」を入力し

ている際のディスプレイ表示を示したものである。(キー 22 に対する) 7 番目のキーストロークの後、データベース中の入力キー順序に一致する単一のオブジェクトは、望ましい入力フレーズそのものである。したがって、そのポイントにおける順序が完全な語やフレーズに対応していない場合であっても、最も頻繁にマッチングする語幹(このケースでは「ありがとう」)がディスプレイ表示される。さらに、データベース中にただ 1 つだけ潜在的に一致するオブジェクトが存在し、このオブジェクトの現在の位置にあるかなが濁点を含むため、Diacritic のキー 68 が作動させられない場合であっても、その仮名は濁点と共に表示される。したがって、この場合、Diacritic キー 68 の作動は任意であり、例 2 は Diacritic キー 68 を作動させずに、「ご」又は「ざ」を示す場合を示す。1 つの語幹が 7 番目で示され、入力フレーズ(句)が 11 のキーストロークで完成するまで、キーストロークが順序の 10 番目であっても、語幹は 7 番目でディスプレイ表示される。

【0170】例 3 は、好ましい実施形態において、選択キー 60 及び変換キー 62 に関連付けられた様々な機能を説明するものである。キー 22 とキー 25 に対する最初の 2 つのキーストロークの後で、最も頻度高く一致する読み方が「この」として決定される。次の、選択キー 60 に対するキーストロークは、現在の単語オブジェクトのデータキー順序の終点を示し、そして、点線の下線を実線の下線に偏向する。選択キー 60 を 2 回押すと、2 番目に頻度の高い読み方「くに」を選択する。それに引き続く変換キー 62 に対するキーストロークにより、前記選択され表示された読み方「くに」に最も共通する見出し語を選択する。それに引き続くコンバートキー 62 に対する 2 回目のキーストロークは、単一漢字文字である最小に共通する見出し語、カタカナの語りで選択された読み方を示す順序の最後の見出し語を巡回する。変換キー 62 の引き続くキーストロークは、最初に表示されたひらがな、見出し語の順序における最初のテキスト解釈であり読み方と同一であるひらがなに戻る。さらなる追加の変換キー 62 に対するキーストロークは、巡回を繰り返させ、再度最も頻度の高い漢字解釈を示す。この後に従って、選択キー 60 を押すと、現在選択されている読み方「くに」の表示に復帰する。選択キー 60 を 2 度押すと、次に(3 番目に)頻度の高い読み方「かね」に移行する。次引き続く変換キー 62 に対するキーストロークは、読み方に関連付けられた最初の 2 つ(最も共通)の見出し語である単一漢字を通して巡回する。最後の変換キー 62 のプレスアンドホールドは、見出し語リストを逆戻りに巡回し、以前に表示した漢字を再選択する。

【0171】例 4 は、どのように自動活用が機能することができるかについて、図示している。最後の結果は、「送ってください」という意味である。ルートワー

ドは、「送」で、送るという意味である。次の 2 つの仮名である、「って」は、連続的な(continuative)ケースを述べたものである。最後の 4 つの仮名、「ください」は、「お願いします」を表している。このワードは、その完全なデータベースで格納されることができ、そして、その動詞の分類の接尾辞ツリーのすべての仮名と動詞の分類を特定するベースツリーのルートワードとして形成されることができ、あるいは、ベースツリーのルートワード、中間接尾辞ツリーの次の 2 つの仮名、そして、最後の接尾辞ツリーの最後の 4 つの仮名、から形成されることができる。

【0172】上述された小型キーボードによる日本語テキスト入力システムによれば、コンピュータ及びシステムに含まれるその他の装置のサイズを減少させる。キーの数を減らすことによって、装置をユーザーが片手で保持することができるように構成することができ、他の手で操作することを可能にする。明らかにされたシステムは、正確で高速なテキスト入力ができることから、携帯電話、PDA、双方向ページャ、若しくは他の小型電子装置での使用に有効である。このシステムは、タッチスクリーンを有する装置若しくは限られたディスプレイスクリーン領域と限られた数のキーしか有しない装置に適用されることで効率と単純化の両方に寄与することができる。この発明のシステムは、日本語音節からなるかなの形で入力を生成し、入力された仮名を意図する漢字若しくは仮名の読みに対応する他のテキストに変換する工程を統合したものである。さらに、この発明のシステムは、データベースに、非常にコンパクトで最小の処理容量で処理できるように、システムの運転に必要とされる情報を格納する方法を提供する。

【0173】なお、当業者であれば、キーボード配列の設計や基本的なデータベースの設計については、この発明の要旨を変更しない範囲で多少の変形が可能である。また、付加されたクレームの範囲内において、ここに記述したように具体的に実施可能である。

【0174】

【発明の効果】上述された小型キーボードによる日本語テキスト入力システムによれば、コンピュータ及びシステムに含まれるその他の装置のサイズを減少させる。キーの数を減らすことによって、装置をユーザーが片手で保持することができるように構成することができ、他の手で操作することを可能にする。明らかにされたシステムは、正確で高速なテキスト入力ができることから、携帯電話、PDA、双方向ページャ、若しくは他の小型電子装置での使用に有効である。このシステムは、タッチスクリーンを有する装置若しくは限られたディスプレイスクリーン領域と限られた数のキーしか有しない装置に適用されることで効率と単純化の両方に寄与することができる。この発明のシステムは、日本語音節からなるかなの形で入力を生成し、入力された仮名を意図する漢字

若しくは仮名の読みに対応する他のテキストに変換する工程を統合したものである。さらに、この発明のシステムは、データベースに、非常にコンパクトで最小の処理容量で処理できるように、システムの運転に必要とされる情報を格納する方法を提供する。

【図面の簡単な説明】

【図 1】本発明の実施形態に係る小型化されたキーボードの曖昧性を解消しているシステムを組み込んだ携帯電話の概略図。

【図 2】本発明の実施形態に係る限界の、または、ディスプレイの能力なしによる小型化されたキーボードシステムを組み込んだ携帯電話のキーパッドの概略図。

【図 3】本発明の実施形態に係る入力された一対の曖昧でない 2 入力の方法におけるキーストロークの第 1 として、キー 2 の後にディスプレイに表示するディスプレイ能力にを有する、キーボードシステムを組み込んだ携帯電話のキーパッドの概略図。

【図 4】本発明の実施形態に係る図 1 のシステムの曖昧性を解消して、小型化したキーボードのソフトウェアブロックダイアグラムの図である。

【図 5】本発明の実施形態に係る一対の曖昧でない 2 ストローク方法におけるキーストロークの最初のキーストロークの前に表示されつつあることを示す小型化したキーボードを組み込んだポータブルコンピュータのタッチスクリーンの概略図。

【図 6】本発明の実施形態に係るキー操作の一対のキー入力の最初として、音節“か”、“き”、“く”、“け”及び“こ”に相当するキー動作によって、表示されつつあることを示す小型化したキーボードを組み込んだ図 5 のタッチスクリーンの概略図。

【図 7】本発明の実施形態に係る日本語のための、小型化されたシステムの曖昧性を解消するキーボードのためのソフトウェアの曖昧性を解消する word-level の流れを示す図。

【図 8】本発明の実施形態に係る日本語のための、小型化されたシステムの曖昧性を解消するキーボードのためのソフトウェアの曖昧性を解消する word-level の流れを示す図。

【図 9】本発明の実施形態に係る明確に入力された文字で、すでに入力されたキーのための再変換モードのフローチャートを示す図。

【図 10】本発明の実施形態に係る小型化したキーボードを組み込んだポータブルコンピュータのタッチスクリーンの概略図。

【図 11】本発明の実施形態に係る 9 つのキーを有する、小型化したキーボードを組み込んだポータブルコンピュータのタッチスクリーンの概略図。

【図 12】本発明の実施形態に係るキーへの記号の物理的結合と物理的キーに現れるキャラクタの追加の強調された変化を含む論理的結合の例を持つキーとの比較を表

す図。

【図 13】本発明の実施形態に係る論理的記号をキーインデックスに結び付ける表を示す図。

【図 14】本発明の実施形態に係る語彙モジュールのツリーのノードにおいてデータの好ましい内部の処理を表す図。

【図 15】本発明の実施形態に係る読み方のテキストオブジェクトを生成するための、第 1 の方法の語彙に関する構成を表す図。

【図 16】本発明の実施形態に係る読み方の 4 つの異なるタイプの見出し語のテキストオブジェクトを造るために使われる第 2 の命令の語彙に関する構成を表す図。

【図 17】本発明の好ましい実施形態に係る読み方の 4 つの異なるタイプの命令の内の 2 つの見出し語のテキストオブジェクトを造るために使われる第 2 の命令の語彙に関する構成を表す図。

【図 18】本発明の実施形態に係る有効な自動活用の準備のための構造に加える情報を表している図。

【図 19】本発明の実施形態に係るノードの構造において可能な内部のデータアイテムの 4 つの実施の形態を表す図。

【図 20】本発明の実施形態に係る圧縮されない語彙モジュールの好ましいツリー構造を表す図。

【図 21】本発明の好ましい実施形態に係るオブジェクトリスト語彙モジュールから取り出される途中のオブジェクトの中間の記憶装置のための実施の形態を表す図。

【図 22】本発明の別の実施形態に係るキー入力のリストを与えられた語彙モジュールから、テキストオブジェクトを取り出すためのソフトウェアプロセスの流れを表す図。

【図 23】本発明の別の実施形態に係るキー入力のリストを与えられた語彙モジュールから、テキストオブジェクトを取り出すためのソフトウェアプロセスの流れを表す図。

【図 24】本発明の別の実施形態に係るキー入力のリストを与えられた語彙モジュールから、テキストオブジェクトを取り出すためのソフトウェアプロセスの流れを表す図。

【図 25】本発明の別の実施形態に係るキー入力のリストを与えられた語彙モジュールから、テキストオブジェクトを取り出すためのソフトウェアプロセスの流れを表す図。

【図 26】本発明の別の実施形態に係るキー入力のリストを与えられた語彙モジュールから、テキストオブジェクトを取り出すためのソフトウェアプロセスの流れを表す図。

【図 27】本発明の実施形態に係る 1 つのキー入力を与えられた語彙モジュールのツリー構造を通過し、そして、オブジェクトリストの状態を変更するためのソフトウェアプロセスの流れを表す図。

【図28】本発明の実施形態に係る圧縮された語彙モジュールを展開するためのソフトウェアプロセスの流れを表す図。

【図29】本発明の実施形態に係る語彙モジュールのツリーデータ構造を展開するためのソフトウェアプロセスの流れを表す図。

【図30】本発明の実施形態に係る1秒当たり設置ノードのためのソフトウェアプロセスのあるノードとの比較に最も大きい重複度を持つ語彙モジュールのツリーの流れを表す図。

【図31】本発明の実施形態に係る語彙モジュールにおいてツリーの2つのノードの間の重複を計算するためのソフトウェアプロセスの流れを表す図。

【図32】本発明の実施形態に係る音節“か”、“き”、“く”、“け”及び、“こ”とあいまいに関連している、キー上の一連の3つの連続したキーストロークのためのデータベースの部分的な内容の流れを表す図。

【図33】本発明の実施形態に係る音節“か”、“き”、“く”、“け”及び、“こ”とあいまいに関連している、キー上の一連の3つの連続したキーストロークのためのデータベースの部分的な内容の流れを表す図。

【図34】本発明の実施形態に係る、図1に示されるシステムのテキストの内容をディスプレイに示するとき、テキストの入力中に、一連のキー上での、各キー入力に続く入力操作のシステム制御の4つの代表的な実施*

*の形態を表す図。

【図35】本発明の実施形態に係る基本的な日本語の音節表を示す図。

【図36】本発明の実施形態に係る発音区別に用いる、追加の日本語の音節表を示す図。

【図37】本発明の実施形態に係る口蓋音化された母音が付加された日本語の音節表を示す図。

【図38】本発明の実施形態に係る日本語の音節表に関する分類表を示す図。

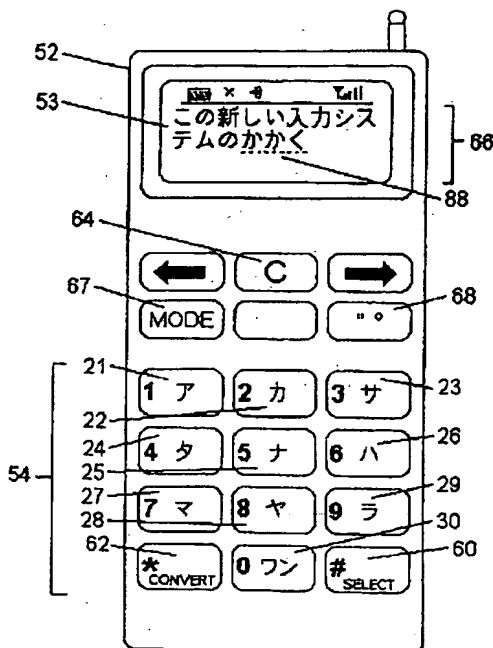
10 【図39】本発明の実施形態に係る日本語の音節表に関する分類表を示す図。

【図40】本発明の実施形態に係る日本語の音節表に関する互い違いの(Alternate)分類表を示す図。

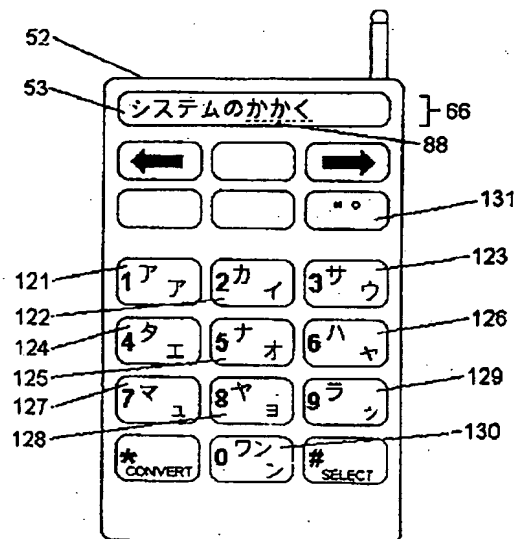
【符号の説明】

- 21～30…キー
- 52…携帯型電話
- 53…ディスプレイ
- 54…キーボード
- 60…変換キー
- 62…選択キー
- 64…クリアキー
- 66…テキスト領域
- 67…モードキー
- 68…区別的発音符号キー
- 77…選択リスト領域
- 88…挿入ポイント

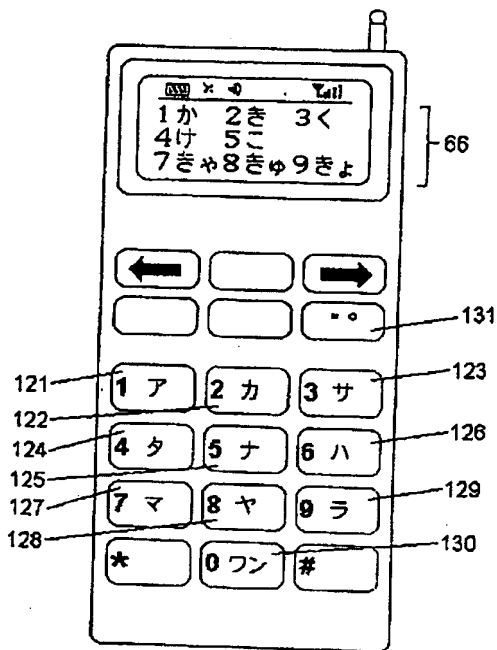
【図1】



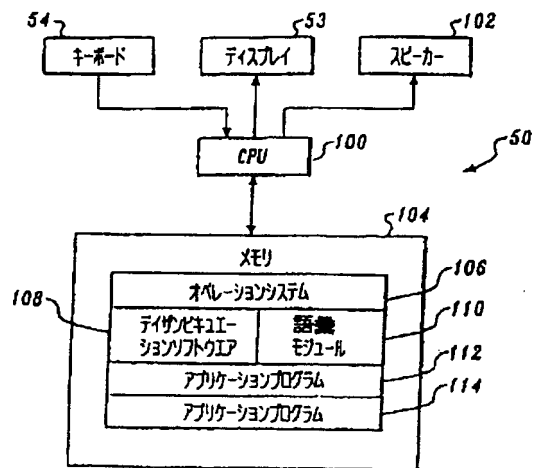
【図2】



【図3】

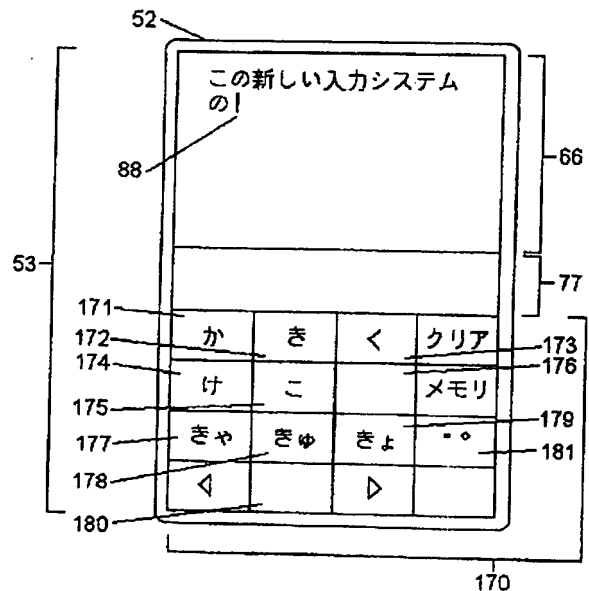
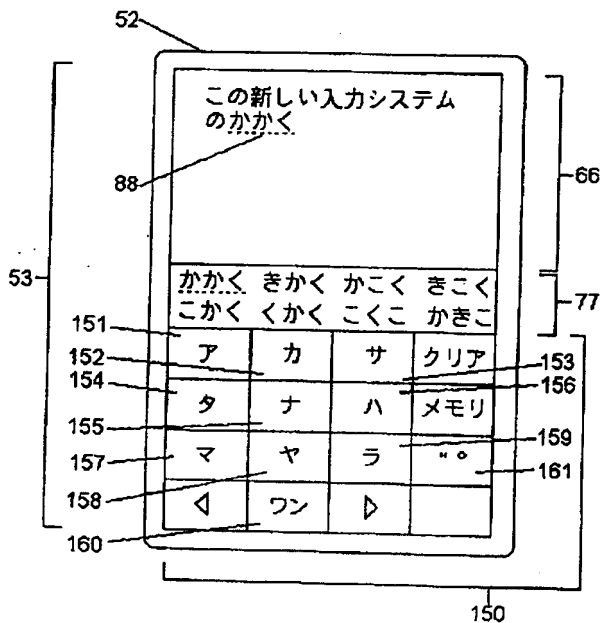


【図4】



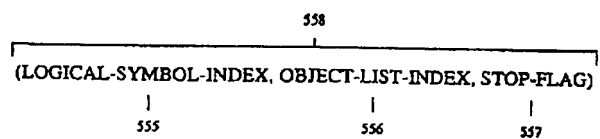
【図6】

【図5】

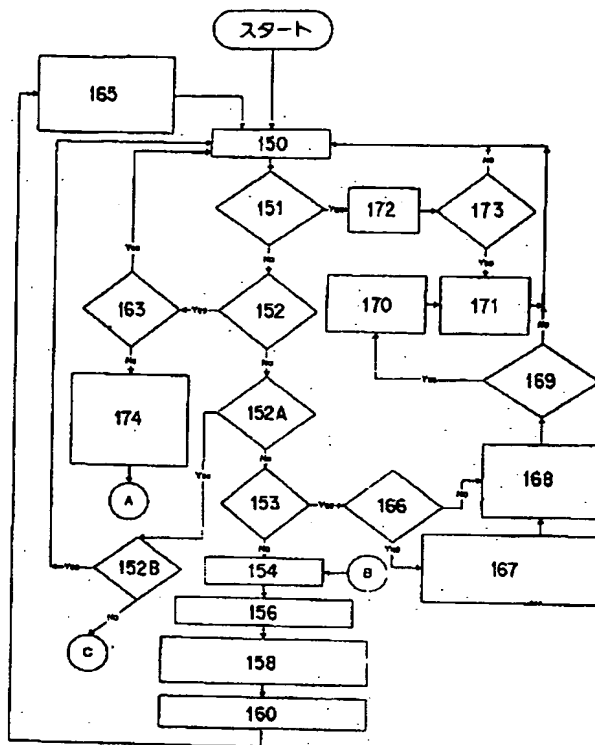


【図15】

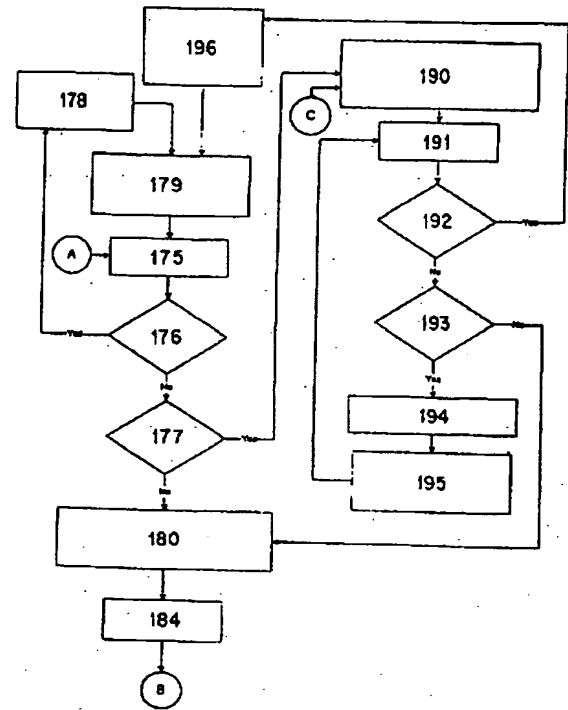
第1の指示のための指示の
記号論 (Semantics)



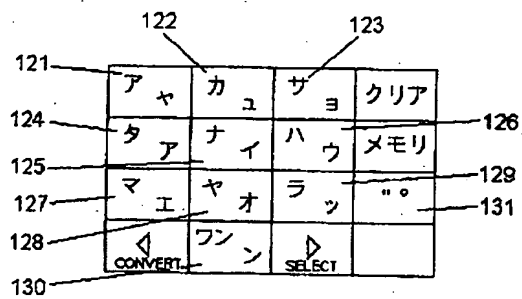
【図7】



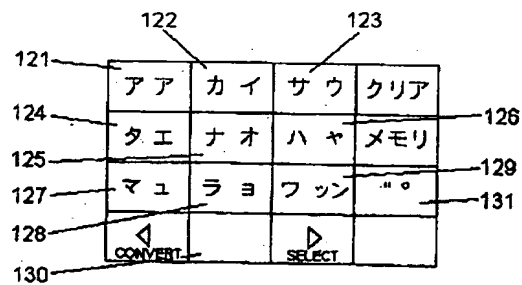
【図8】



【図10】

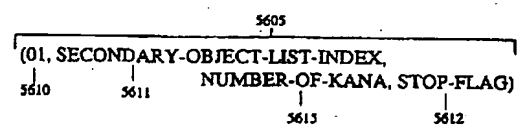
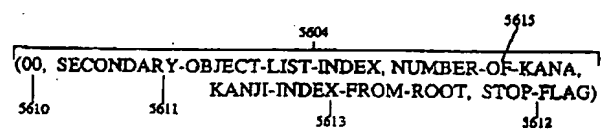


【図11】

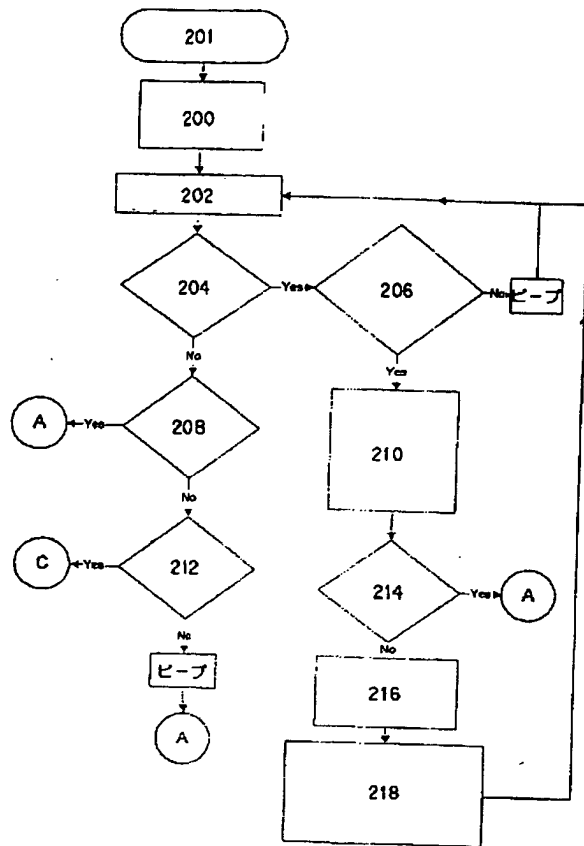


【図17】

第2の指示のための代わりの
指示の記号論 (Semantics)



【図 9】

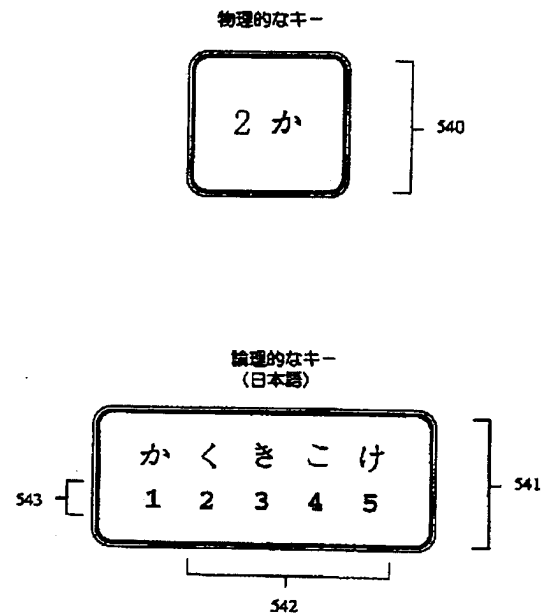


【図 13】

日本語のための
論理的記号
インデックス表

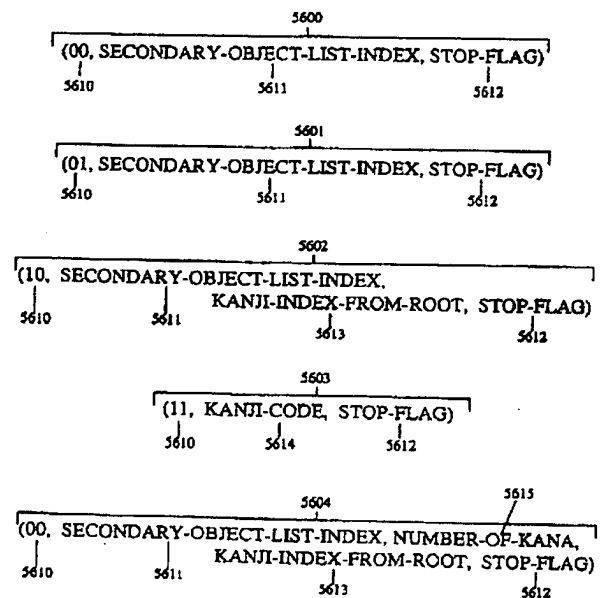
	1	2	3	4	5	6	7	8	9	10
1	い	う	お	あ	え	い	と	あ	せ	う
2	か	く	き	こ	け					
3	し	さ	す	せ	そ					
4	て	つ	と	っ	ち	た				
5	な	に	の	ね	ぬ					
6	は	ひ	ふ	ほ	へ					
7	ま	み	も	め	む					
8	よ	ゆ	よ	や	や	ゆ				
9	り	る	ら	れ	ろ					
10	ん	-	わ	を						
11	"	。								

【図 12】

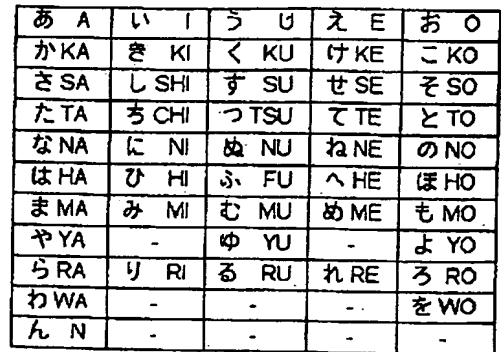


【図 16】

第2の指示のための指示の
記号論 (Semantics)



【図 35】



【図 19】

(SUFFIX CLASS, FREQUENCY, STOP-FLAG)

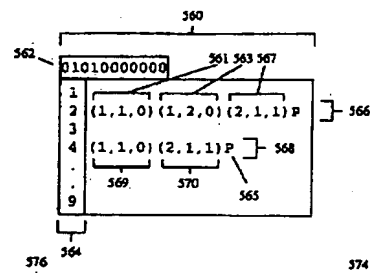
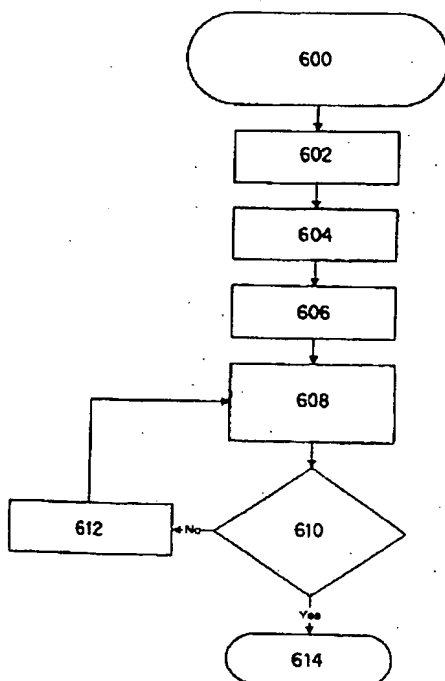


Figure 1 illustrates the structure of the Huffman tree and the associated data blocks. The diagram shows three memory blocks, each containing a 10-bit binary string and a list of 5 elements. The blocks are labeled 577, 571, and 582. The binary strings are 01010000100, 01010000000, and 01010000100 respectively. The lists of elements are:

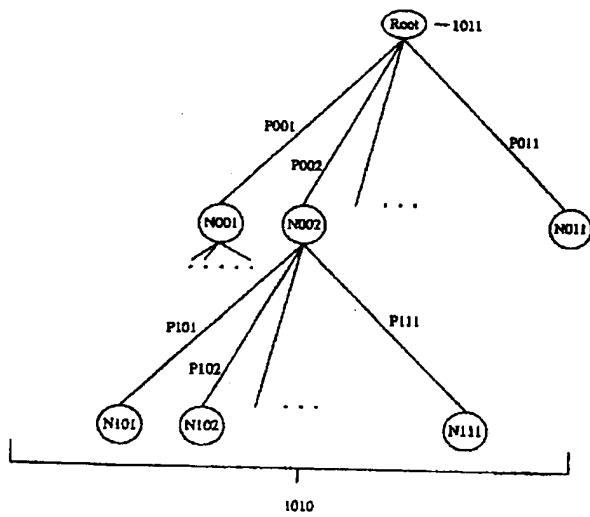
- Block 577: 1: (1,1,0) (1,2,0) (2,1,1) P; 2: (1,1,1) P; 3: (1,1,1) P; 4: (1,1,1) P; 5: (1,1,1) P
- Block 571: 1: (1,1,0) (1,2,0) (2,1,1) P; 2: (1,1,1) P; 3: (1,1,0) (2,1,1) P; 4: (1,1,1) P; 5: (1,1,1) P
- Block 582: 1: (1,1,0) (1,2,0) (2,1,1) P; 2: (1,1,1) P; 3: (1,1,0) (2,1,1) P; 4: (1,1,1) P; 5: (1,1,1) P

The lists are grouped into three sets: 577, 571, and 582. The lists are grouped into three sets: 577, 571, and 582.

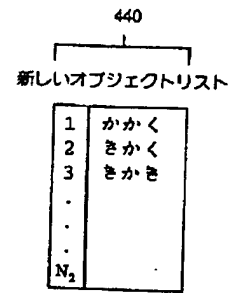
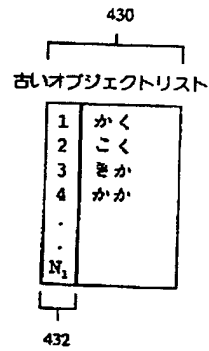


が GA	ぎ GI	ぐ GU	げ GE	ご GO
ざ ZA	じ JI	ず ZU	ぜ ZE	ぞ ZO
だ DA	ぢ JI	づ ZU	で DE	ど DO
ば BA	び BI	ぶ BU	べ BE	ぼ BO
ぱ PA	ぴ PI	ぷ PU	ぺ PE	ぽ PO

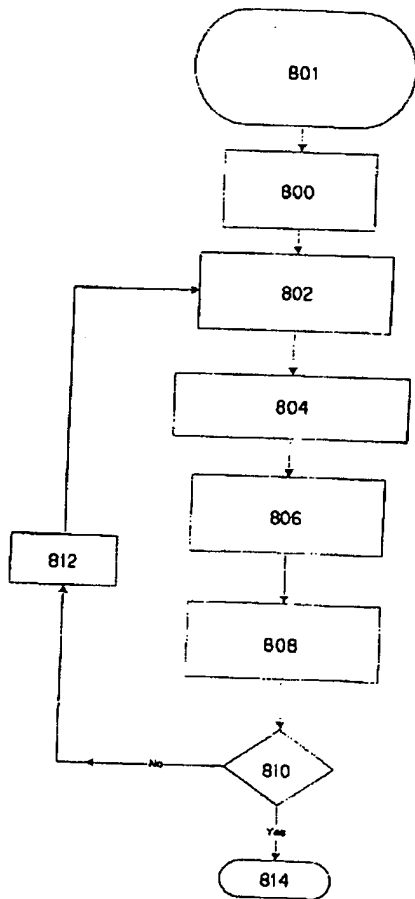
【図 20】



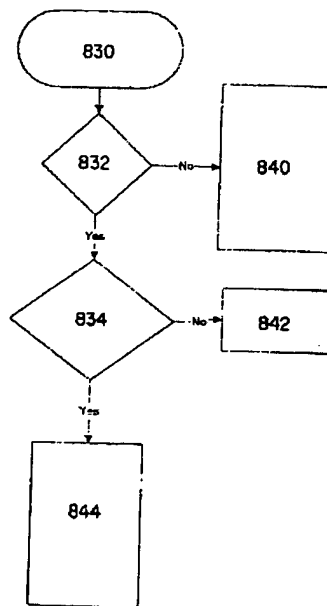
【図 21】



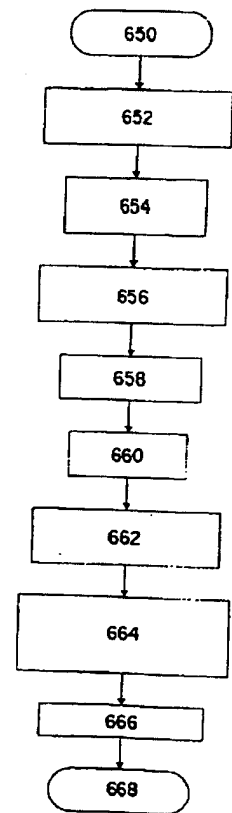
【図 23】



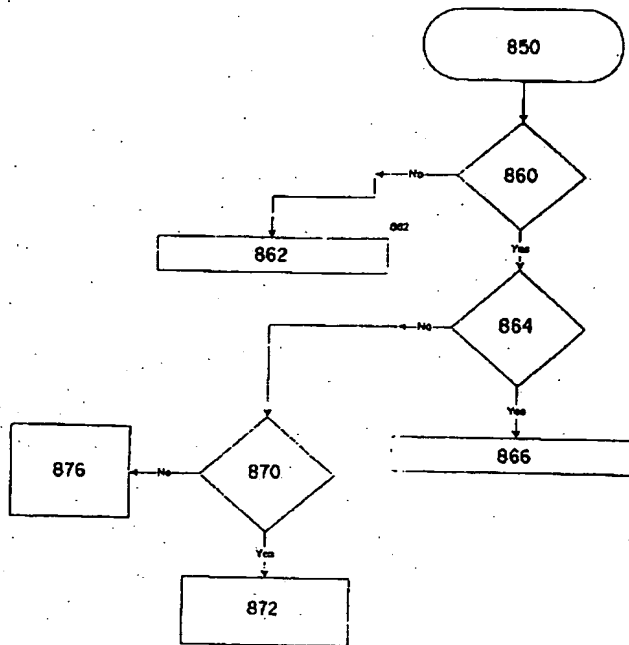
【図 24】



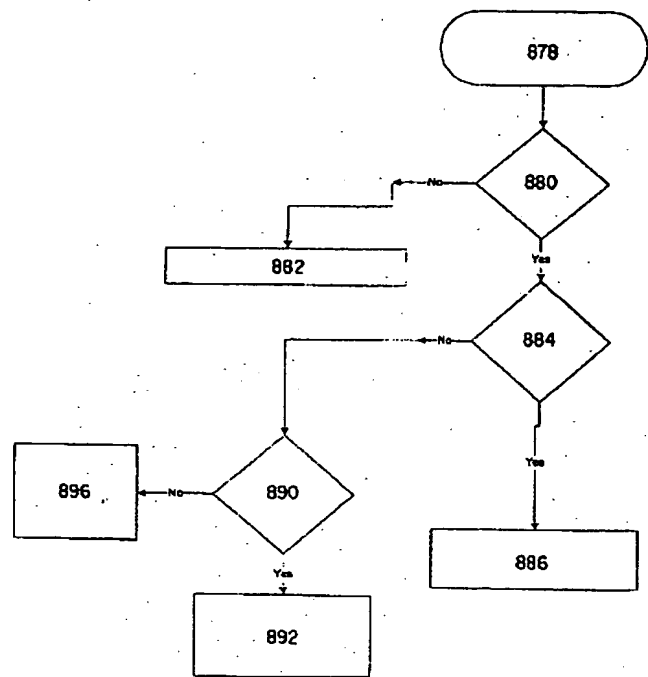
【図 28】



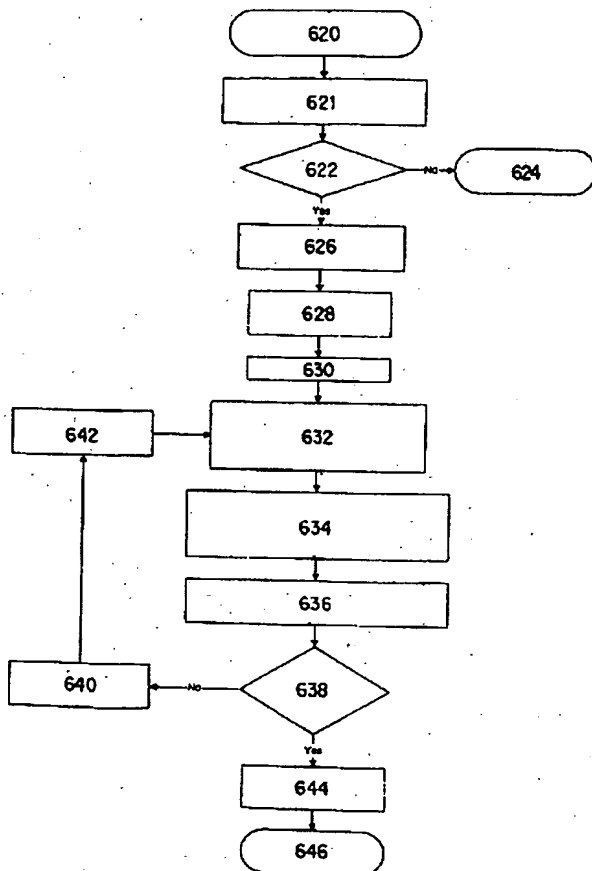
【図25】



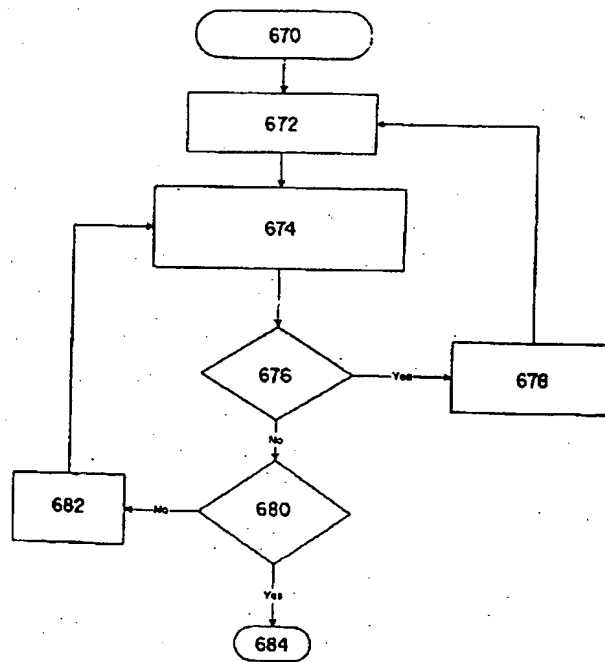
【図26】



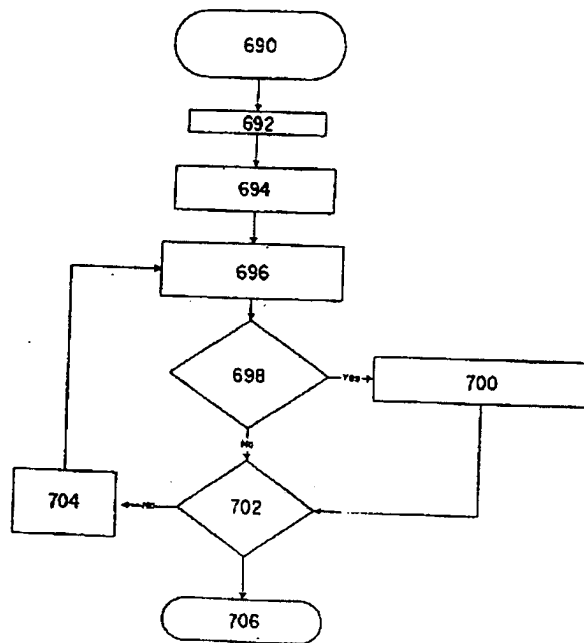
【図27】



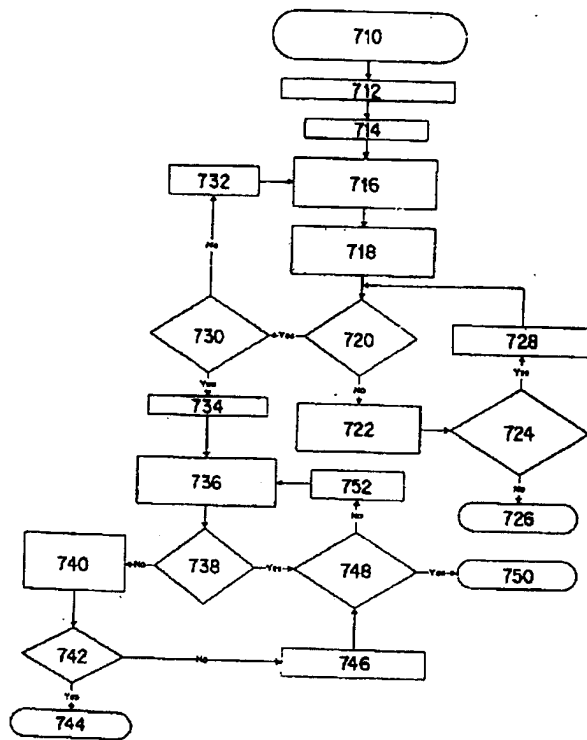
【図29】



【図30】



【図31】



【図32】

レベル1 2 か	レベル2 2 か	レベル3 2 か
Y1: Y0 + か M1: M0 + 化 M2: M0 + 録 M3: M0 + 火 M4: M0 + 科 M17: M0 + 価 M23: M0 + 稼 M24: M0 + 箇 M25: M0 + 描 M26: M0 + 何 M27: M0 + 欠 M28: M0 + 書 M29: M0 + 質 Y2: Y0 + き M1: M0 + 期 M2: M0 + 器 M3: M0 + 提 M4: M0 + 倉 M5: M0 + 気	Y1: Y1 + < {かく} M1: M0 + 各 M2: M0 + 核 M3: M28 + < {書く} M4: M0 + 開 M5: M0 + 画 M6: M25 + < {描く} M7: M0 + 格 M8: M0 + 角 M9: M27 + < {欠く} M10: M0 + 確 M11: M0 + 振 M12: M0 + 殺 M13: M0 + 獲 M14: M29 + [M3] {質来} M15: M0 + 割 M16: M0 + 掘 Y2: Y4 + < {にく} M1: M0 + 固 M2: M0 + 石	Y1: Y16 + < {かかく} M1: M2 + [M7] {備格} Y2: Y11 + < {きかく} M1: M5 + [M5] {企画} M2: M6 + [M7] {規格} Y3: Y17 + < {きこく} M1: M1 + [M1] {帰国} Y9: Y1 + か {かくか} M1: M1 + [M2] {各課} Y13: Y11 + き {きかき} M1: M2 + [M2] {気化器}

【図37】

きゃ KYA	きゅ KYU	きょ KYO
ぎゃ GYA	ぎゅ GYU	ぎょ GYO
しゃ SHA	しゅ SHU	しょ SHO
じゃ JA	じゅ JU	じょ JO
ちゃ CHA	ちゅ CHU	ちょ CHO
にゃ NYA	にゅ NYU	にょ NYO
ひゃ HYA	ひゅ HYU	ひょ HYO
びゃ BYA	びゅ BYU	びょ BYO
ぴゃ PYA	ぴゅ PYU	ぴょ PYO
みゃ MYA	みゅ MYU	みょ MYO
りゃ RYA	りゅ RYU	りょ RYO

【図33】

レベル1 2か	レベル2 2か	レベル3 2か
Y2: Y0 + き (cor.) M8: M0 + 帰 M7: M0 + 企 M8: M0 + 選 M9: M0 + 規 M10: M0 + 聞 Y3: Y0 + く M1: M0 + 九 M2: M0 + 区 M3: M0 + 来 Y4: Y0 + こ M1: M0 + 個 M2: M0 + 子 M3: M0 + 小 Y5: Y0 + け M1: M0 + 気 M2: M0 + 毛 M3: M0 + 消	Y3: Y2 + き (きき) M1: M3 + [M2] (機弱) M2: M4 + [M3] (危機) Y4: Y4 + こ (ここ) M1: M1 + 々 (個々) Y11: Y2 + か (きか) M1: M6 + [M1] (帰化) M2: M5 + [M1] (気化) M3: M10 + か (聞か) M4: M8 + [M26] (機何) M5: M7 + P (企口) M6: M9 + P (規口) Y16: Y1 + か (かか) M1: M28 + か (書か) M2: M17 + P (節口) Y17: Y2 + こ (きこ) M1: M8 + P (帰口)	

【図34】

例1

挿入部外図における
表示された入力外

キ-21の後: お
 キ-25の後: あの
 キ-22の後: おなか
 キ-68の後: うなぎ
 キ-21の後: おねがい
 キ-62の後: お願
 キ-23の後: お願
 キ-27の後: お願
 キ-23の後: お願
 キ-60の後: お願

例3

挿入部外図における
表示された入力外

キ-21の後: お
 キ-29の後: いる
 キ-22の後: いらく
 キ-68の後: ありが
 キ-24の後: いろいろ
 キ-21の後: ありがとう
 キ-22の後: ありがとう
 キ-23の後: ありがとう
 キ-21の後: ありがとう
 キ-27の後: ありがとう
 キ-23の後: ありがとう

例2

挿入部外図における
表示された入力外

キ-22の後: か
 キ-25の後: この
 キ-60の後: この
 キ-60の後: くに
 キ-62の後: 国
 キ-62の後: 邦
 キ-62の後: くに
 キ-62の後: 国
 キ-60の後: くに
 キ-60の後: かね
 キ-62の後: 金
 キ-62の後: 鐘
 キ-の押し付け
 の後62: 金

例4

挿入部外図における
表示された入力外

キ-21の後: お
 キ-22の後: いが
 キ-24の後: あきた
 キ-24の後: あかた
 キ-22の後: あかた
 キ-24の後: あかた
 キ-68の後: うきた
 キ-23の後: うきた
 キ-21の後: うきた
 キ-62の後: うきた

【図38】

Key 1 \ Key 2	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ツ	ン
ア	あ A	い I	う U	え E	お O					
カ	か KA	き KI	く KU	け KE	こ KO	きゃ KYA	きゅ KYU	きょ KYO		
サ	さ SA	し SHI	す SU	せ SE	そ SO	しゃ SHA	しゅ SHU	しょ SHO		
タ	た TA	ち CHI	つ TSU	て TE	と TO	ちゃ CHA	ちゅ CHU	ちょ CHO	っ(TSU)	
ナ	な NA	に NI	ぬ NU	ね NE	の NO	にゃ NYA	にゅ NYU	にょ NYO		
ハ	は HA	ひ HI	ふ FU	へ HE	ほ HO	ひゃ HYA	ひゅ HYU	ひょ HYO		
マ	ま MA	み MI	む MU	め ME	も MO	みゃ MYA	みゅ MYU	みょ MYO		
ヤ	や YA		ゆ YU		よ YO	ゃ (YA)	ゅ (YU)	ょ (YO)		
ラ	ら RA	り RI	る RU	れ RE	ろ RO	りゃ RYA	りゅ RYU	りょ RYO		
ワン	わ WA			を WO						ん N

【図39】

Key 1 \ Key 2	1	2	3	4	5	6	7	8	9	0
ア	あ A	い I	う U	え E	お O	め (M)	い (I)	う (U)	え (E)	お (O)
カ	か KA	き KI	く KU	け KE	こ KO		きゃ KYA	きゅ KYU	きょ KYO	
サ	さ SA	し SHI	す SU	せ SE	そ SO		しゃ SHA	しゅ SHU	しょ SHO	
タ	た TA	ち CHI	つ TSU	て TE	と TO	っ(TSU)	ちゃ CHA	ちゅ CHU	ちょ CHO	
ナ	な NA	に NI	ぬ NU	ね NE	の NO		にゃ NYA	にゅ NYU	にょ NYO	
ハ	は HA	ひ HI	ふ FU	へ HE	ほ HO		ひゃ HYA	ひゅ HYU	ひょ HYO	
マ	ま MA	み MI	む MU	め ME	も MO		みゃ MYA	みゅ MYU	みょ MYO	
ヤ	や YA		ゆ YU		よ YO		ゃ (YA)	ゅ (YU)	ょ (YO)	
ラ	ら RA	り RI	る RU	れ RE	ろ RO		りゃ RYA	りゅ RYU	りょ RYO	
ワン	わ WA			を WO		ん N			ー(長母音)	

【図40】

Key 1 \ Key 2	ア	イ	ウ	エ	オ	ヤ	ユ	ヨ	ン
ア	あ A	い I	う U	え E	お O	や YA	ゆ YU	よ YO	
カ	か KA	き KI	く KU	け KE	こ KO	きゃ KYA	きゅ KYU	きょ KYO	
サ	さ SA	し SHI	す SU	せ SE	そ SO	しゃ SHA	しゅ SHU	しょ SHO	
タ	た TA	ち CHI	つ TSU	て TE	と TO	ちゃ CHA	ちゅ CHU	ちょ CHO	っ(TSU)
ナ	な NA	に NI	ぬ NU	ね NE	の NO	にゃ NYA	にゅ NYU	にょ NYO	
ハ	は HA	ひ HI	ふ FU	へ HE	ほ HO	ひゃ HYA	ひゅ HYU	ひょ HYO	
マ	ま MA	み MI	む MU	め ME	も MO	みゃ MYA	みゅ MYU	みょ MYO	
ラ	ら RA	り RI	る RU	れ RE	ろ RO	りゃ RYA	りゅ RYU	りょ RYO	
ワン	わ WA			を WO					ん N

フロントページの続き

(51) Int. Cl.⁷

識別記号

F I

ターマコード(参考)

G 0 6 F 17/22

5 0 4

G 0 6 F 17/22

5 0 4 A

H 0 4 M 1/00

H 0 4 M 1/00

W

1/725

1/725

(72) 発明者 エドワード・ピー・フリンケム
 アメリカ合衆国、ワシントン州 98119、
 シアトル、ダブリュ・キニア・プレイス
 118

(72) 発明者 イーサン・アール・ブラッドフォード
 アメリカ合衆国、ワシントン州 98103、
 シアトル、エヌ・サーティーフス・ス
 トリート 1412

(72) 発明者 ダイジュ・マツオ
 アメリカ合衆国、ワシントン州 98133、
 シアトル、ナンバー 606、メリディア
 ン・アベニュー・エヌ 10306

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK (USPTO)